The Du Fort and Frankel finite difference scheme applied to and adapted for a class of finance problems

by

Abraham Bouwer

Submitted in partial fulfillment of the requirements for the degree

Magister Scientiae

in the Faculty of Natural & Agricultural Sciences
University of Pretoria
Pretoria

April 2008

DECLARATION

I, the undersigned, hereby declare that the dissertation submitted herewith for the degree Magister Scientiae to the University of Pretoria contains my own, independent work and has not been submitted for any degree at any other university.

Signature:

Name: Abraham Bouwer

Date: 2008–04–30

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Title	The Du Fort and Frankel finite difference scheme applied to and adapted for a class of finance problems
Name	Abraham Bouwer
Supervisor	Prof E Maré
Department	Mathematics and Applied Mathematics
Degree	Magister Scientiae

Summary

We consider the finite difference method applied to a class of financial problems. Specifically, we investigate the properties of the Du Fort and Frankel finite difference scheme and experiment with adaptations of the scheme to improve on its consistency properties.

The Du Fort and Frankel finite difference scheme is applied to a number of problems that frequently occur in finance. We specifically investigate problems associated with jumps, discontinuous behavior, free boundary problems and multi dimensionality. In each case we consider adaptations to the Du Fort and Frankel scheme in order to produce reliable results.

Contents

List of Tables

List of Figures

Glossary

Boundary condition. The finite difference scheme estimates a solution for the differential equation over a discrete interval of the spatial variable. The solution of the partial differential equation must be specified for the upper and lower values of the interval. This specification is known as the (upper and lower) boundary conditions.

Black and Scholes partial differential equation. A partial differential equation derived by Black and Scholes [1973] that describes the arbitrage free price of a contingent claim.

Consistency of a finite difference scheme. A finite difference scheme is considered to be consistent with the partial differential equation if the truncation error tend to 0 when the temporal and spatial step sized trend to zero.

Convection term. The first derivative of the option price with respect to the spatial variable(s).

Crank and Nicolson's finite difference scheme. An unconditionally stable implicit finite difference scheme that is second order accurate in both the spatial and temporal dimension. First published by Crank and Nicolson [1947].

Diffusion term. The diffusion term is the second derivative of the price of the contingent claim with respect to the spatial variable(s).

Douglas scheme. An implicit finite difference scheme that utilizes explicit and implicit difference equation in an optimized way in order to achieve fourth order accuracy in the spatial direction and second order accuracy in the temporal direction.

Du Fort and Frankel's finite difference scheme. An unconditionally stable explicit finite difference scheme that is second order accurate in both the spatial and temporal dimensions. First published by Du Fort and Frankel [1953]. The scheme is only conditionally consistent with the partial differential equation.

Explicit finite difference scheme. An explicit finite difference scheme is a finite difference scheme that has an unknown vector $\mathbf{v}$ in the matrix relation

$$\mathbf{v} = \mathbf{M}\mathbf{x},$$

where $\mathbf{M}$ is a known square matrix and $\mathbf{x}$ is a known vector.

Finite difference scheme. A finite difference scheme is a numerical method where by differential equations (or partial differential) equations are solved by estimating derivatives discretely by making use of first differences.

Finite difference mesh.

Heat equation. A partial differential equation that describes the flow of heat over time in a linear conductor.

Implicit finite difference scheme. An implicit finite difference scheme is a finite difference scheme that has an unknown vector $\mathbf{v}$ in the matrix relation

$$\mathbf{M}\mathbf{v} = \mathbf{x},$$

where $\mathbf{M}$ is a known square matrix and $\mathbf{x}$ is a known vector.

Local truncation error of a finite difference scheme. The local truncation error measures by how much the approximating difference equation does not satisfy the original partial differential equation at specified mesh points.

Mesh. See finite difference mesh.

Mesh point. A pair of temporal and spatial variables such that the values for the variables are integer multiples of time-step sizes and spatial-step sizes plus the minimum discretised temporal value and minimum discretised spatial value.

Stability of a finite difference scheme. A finite difference scheme is considered stable if errors remain bounded.

Notation

$\mathbf{b^i}$	Vector containing boundary conditions at time-step i.
$\mathbf{e^i}$	Vector containing rounding errors at timestep i.
f_j^i	An approximation of function $F(\tau, S_\tau)$
$\mathbf{f^i}$	A vector of values for f at time-step i.
F_j^i	Shorthand for $f(q\chi + ik, s_\chi + jh)$.
h	Length of a discretised spatial interval.
i	A reference to the discrete time step.
j	A reference to the discrete spatial step.
k	Length of a discretised temporal interval.
q	Discretised temporal variable.
q_χ	Initial discretised time.
q_ψ	Terminal discretised time.
r	Risk free interest rate.
s	Discretised share price.
s_χ	The discretised share price at the lower spatial boundary.
s_ψ	The discretised share price at the upper spatial boundary.
t	A time in the interval $[t_0, T]$.
v_j^i	Discrete approximation for $V(\zeta, \Xi_\zeta)$.
v_j^i	Shorthand for $v(\nu_\chi + i\kappa, \xi_\chi + j\eta)$.
$\mathbf{v^i}$	Vector of values for v at time-step i.

A, B, C, D	Also variations such as $\overline{A}$, $\dot{A}$ etc. Coefficients for mesh points $(i-1, j+1)$, $(i-1, j)$, $(i-1, j-1)$ and $(i-2, j)$ respectively. Variation depends on the scheme that is used.
F_t	The price of a contingent claim on underlying share price S_t.
dF_t	The price process for a contingent claim.
M	Number of time steps minus one.
$\mathbf{M}$	A square tri-diagonal matrix.
N	Number of temporal steps minus one.
S_t	Share price at time t.
dS_t	Share price process.
T	Maturity of a contingent claim.
T_j^i	Local truncation error at mesh point (i, j).
$V(\zeta, \Xi_\zeta)$	The function resulting from transforming the Black and Scholes PDE to the heat equation. Equivalent to the price of the contingent claim.
W_t	A Gaussian distributed random number.
dW_t	A Wiener process.
α, β, γ	General variables usually associated with the diffusion, convection and functional terms in the partial differential equation.
$\beta^{\mathbf{i}}$	Vector containing boundary conditions used with the heat equation.
η	Spatial strep size for the discretised heat equation.
ζ	The transformed temporal variable in the heat equation.
κ	Temporal step size for the discretised heat equation.
μ	Drift rate of a share.
ν	Discrete temporal variable for the heat equation.
ξ	Discrete spatial variable for the heat equation.
σ	Volatility of a share.
τ	Temporal variable.
ω	Eigenvalues.

$\Delta(t)$	The number of shares held in portfolio $\Pi(t)$ at time t.
Θ	General variable to distinguish between various finite difference schemes.
Ξ_ς	The spatial variable in the heat equation.
$\Pi(t)$	The value of a portfolio at time t.
$\mathbf{\Sigma}$	Square tri-diagonal matrix used with the heat equation.
$\Phi(S_T)$	Terminal boundary condition for contingent claim F_T.

Chapter 1

Introduction and objectives

1.1 Introductory summary

Finite difference schemes represent an important class of numerical procedures employed in finance. The method was well studied and its shortcomings were well known before the advent of its large scale implementation in finance. Many of these schemes were developed for applications not related to finance. The event that triggered their appearance in finance was arguably the publications of Black and Scholes [1973] and Merton [1973]. These works succeeded in describing a class of financial problems, namely contingent claims as a partial differential equation. The number of analytical solutions of this equation is small in comparison with the total universe of possible solutions, and hence methods to solve it numerically were published shortly after the pioneering works with the publication of papers by Brennan and Schwartz [1978] and others.

The suite of classical schemes were soon amended by alternative schemes mainly with the idea of improving the convergence rate. Most of these schemes are implicit and mainly for the reason that implicit schemes have superior stability characteristics compared to explicit schemes. The use of explicit schemes often manifest in trinomial and binomial trees, which represent some of the most popular pricing mechanisms in finance. However, in their classical form, explicit schemes became somewhat stigmatized and obscure. They are often brushed aside with arguments related to the fact that explicit schemes require many more time steps than their implicit counterparts in order

to converge and are therefore less efficient.

Explicit schemes, despite their somewhat temperamental nature, have a number of positive characteristics which make them deserving of a place amongst front–line pricing techniques. We discuss a few in this document, but mainly focus on the issue of matrix inversion (or related techniques) that are required by all implicit schemes. The additional computational effort associated with matrix inversion or similar techniques that solve a matrix equation of the form

$$\mathbf{M}\mathbf{v} = \mathbf{x},$$

is such that explicit schemes – even though they require generally more time steps – still often outperform their implicit counterparts. We thus adopt *computing effort* rather than *number of grid points* as a measure of efficiency.

The Du Fort and Frankel scheme is the main subject of our research. It is exotic in the sense that it possess a shortcoming that is rare and often neglected in general discussion on the finite difference method. The shortcoming under discussion is the fact that the Du Fort and Frankel scheme is only conditionally consistent with the partial differential equation. Precious little recourse in relation to inconsistency is offered in literature. We resort to experimentation in order to find techniques that offer relief.

This document is structured in two parts. Part I entails a theoretical background study leading to the Du Fort and Frankel scheme. The chapter outline of Part I is discussed in the introduction to Part I. The next part afford a closer study of features of finance problems that are problematic to solve numerically. We customize the Du Fort and Frankel scheme in order to successfully cope with these difficulties. A chapter outline of Part II is discussed in the introduction to Part II.

Our method of research is twofold. We firstly study available literature. A list of the works that are cited throughout the document is attached to this document as a bibliography. Often these sources describe problems that are related but not entirely similar to the ones we study. For this reason our second method of research is experimentation. We mainly used Matlab to test our ideas, and as such a number of procedures' source codes are listed in the appendix to this document.

Many of our findings mimic that of other authors derived for nuanced problems. However, a number of findings are unique or have unique properties which we were unable to find in literature. These include

- the use of Richardson's extrapolation in order to improve consistency properties of the Du Fort and Frankel scheme,
- interpolation of two temporal vectors in order to accurately price contingent claims on underlying securities with dividends by using a two time-step finite difference scheme such as the Du Fort and Frankel scheme, and
- a grid refinement technique that interpolates grid points which are applicable to two step finite difference methods.

This study is not exhaustive, but provides a general insight into the use of the Du Fort and Frankel scheme and its applicability to problems pertaining to finance.

1.2 Objectives of the research

The general theme of this research is to establish whether the Du Fort and Frankel finite difference scheme offers functionality in addition to some of the more established methods pertaining to the pricing of contingent claims. This functionality may manifest in various areas such as computational efficiency, algorithmic simplicity or accuracy.
We provide a gradual and general introduction to the theory finite difference schemes, applied to problems in finance. With this theoretical foundation we analyse the classical finite difference schemes in order to define the properties of truncation error, consistency and stability. These properties form a benchmark with which the Du Fort and Frankel scheme is measured and scrutinized.
With the Du Fort and Frankel scheme defined and its general properties established, we apply it to specific problems in finance. These problems are chosen on the basis that they offer certain challenges to the classical numerical techniques. In each instance the robustness of the Du Fort and Frankel scheme is tested. We conclude with a summary of the strengths and weaknesses of the Du Fort and Frankel scheme.

Part I

A theoretical background for finite difference schemes

Introduction to Part I

Part I deals with theoretical aspects of the finite difference method. We derive the Black and Scholes partial differential equation [Black and Scholes, 1973; Merton, 1973] and present it in two forms, namely the initial value Black and Scholes equation and the heat equation. Although we do some work on the heat equation, our effort focuses primarily on the Black and Scholes equation. Theory on the heat equation is abundant and most writers first do the transformation. Transforming the Black and Scholes equation into the heat equation has several advantages. Although the two equations are analytically equivalent, numerically there are differences [Seydel, 2004], the most important is that the heat equation lacks convection. By first transforming one therefore evades problems associated with convection dominance. Furthermore, the heat equation provides a simplified means to study the characteristics of finite difference schemes.
Part I is structured in the following way:

- In Chapter 2 we derive the Black and Scholes partial differential equation.
- In Chapter 3 we define a framework for finite differences. The classical schemes namely the fully explicit, the fully implicit and the Crank and Nicolson [Crank and Nicolson, 1947] schemes are discussed.
- Chapter 4 derives and discusses some of the characteristics of finite difference schemes that may serve as a framework to compare schemes. These characteristics are truncation error, consistency with the partial differential equation and stability.
- The main scheme under discussion for this document, the Du Fort and Frankel [Du Fort and Frankel, 1953] scheme is introduced in Chapter 5. This is done by highlighting two other schemes that share some of the central ideas of the Du Fort and Frankel scheme, namely second order convergence on the temporal axis and stability. The main consequence of the Du Fort and Frankel scheme, namely inconsistency with the partial differential equation is briefly discussed and compared the the modified alternating directional scheme. The Du Fort and Frankel scheme is well documented for the heat equation but little analysis is available on the Black and Scholes equation. We consequently focus our efforts

on the Black and Scholes equation. The non symmetric nature of schemes with convection terms makes analysis more cumbersome, and here also we can only afford to provide an outline for proving stability.

- Chapter 6 is devoted to two topics which frequently recurs in literature which may impact on the usability of the Du Fort and Frankel scheme. Our intention is to solve the Black and Scholes partial differential equation as opposed to the heat equation. The principle disadvantage of this preference is that convection dominance problems may occur. Certain causes of convection dominance problems are documented and we investigate two of the alleged causes namely averaging in the discretisation of the temporal derivative and central convection differencing. The second part of the chapter deals with the inconsistency of the Du Fort and Frankel scheme, and two techniques are considered to reduce the effect of the inconsistent behavior. These are firstly increasing the number of temporal steps and secondly canceling "inconsistent" error terms by Richardson's extrapolation.

Finite difference theory extends far beyond the boundaries of our discussion. We wish to point out that recent literature describe numerous enhancements on the conventional schemes discussed in this document. Unfortunately the ideas central to these do not fit the ideas of the main topic of this document namely achieving good and general approximations with an explicit scheme. The explicit property of the Du Fort and Frankel scheme makes it desirable for a number of reasons, which will be motivated throughout this document. Some of the alternative ideas or schemes are briefly discussed.
Chawla et al. [2003] makes use of the trapezoidal rule in order to discretise the temporal derivative. The resulting scheme, GTF(α), achieves third order temporal accuracy for linear equations such as the Black and Scholes equation, thereby improving on the Crank and Nicolson scheme. A further enhancement by Chawla and Evans [2005] makes use of a Noumerov discretisation [Noumerov, 1924] in the spatial direction and Simpson type time integration in the temporal direction. The scheme is fourth order accurate in both time and space. The scheme requires transformation into the heat equation (in order to conduct the Noumerov discretisation) and presents pentagonal system of equations (as opposed to the trigonal system of the conventional systems) which requires substantially more resources to compute. The more dimensional case is uncertain. Further work on high order systems was conducted by Linde et al. [2006] in

which 6^th order space discretisation is obtained by making use of seven points in the spatial direction. In this case a second dimension will require 49 points and so forth. It is noted that estimations around the boundary are necessarily of lower order, which may result in the general scheme being less than 6^th order especially when non-linear behavior occurs near the boundary. The crux of the high-order system however manifests in the overlay of two grids in order to obtain super fine spatial steps around the discontinuities surrounding for instance the strike price of an option. We will investigate similar procedures in part II

Chapter 2

The Black and Scholes partial differential equation

2.1 Introduction

In this chapter we derive the Black and Scholes partial differential equation ("BS–PDE"). The subsequent chapters solve two modified versions of the Black and Scholes partial differential equation. The first modification is to convert the terminal value problem of the BS–PDE to an initial value problem. The second modification entails the transformation of the BS–PDE to a simpler diffusion partial differential equation.

2.2 The derivation of the Black and Scholes partial differential equation

We assume an arbitrage free market that trades continuously and that is sufficiently liquid [Björk, 2004].

Consider the dynamics of the value of a portfolio $\Pi(t)$ over time. The portfolio consists of a basket of liquid shares and a single contingent claim on those shares. The number of shares is denoted by $\Delta(t) = (\Delta_1(t), \Delta_2(t), \ldots, \Delta_N(t))$ and their prices by $S(t) = (S_1(t), S_2(t), \ldots, S_N(t))$. The contingent claim is denoted by $F(t, S(t))$. We often parameterise the time dependence of the share prices, i.e. $S_t \equiv S(t)$. The value

of the portfolio will be considered over the ordered time interval $t \in [t_0, T]$ where t_0 denotes the valuation date of the contingent claim and T denotes the maturity date.

We assume the share prices follow geometric Brownian motion,

$$dS_t = \mu S_t dt + \sigma S_t dW(t), \tag{2.1}$$

where

$$\begin{aligned} \mu = (\mu_1, \mu_2, \ldots, \mu_N) &= \text{the drift coefficients of the share prices,} \\ \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N) &= \text{the volatility coefficients of the share prices, and} \\ dW(t) = (dW_1(t), dW_2(t), \ldots, dW_N(t)) &= \text{Wiener processes.} \end{aligned}$$

The share price returns may be correlated thus we qualify the various Wiener processes by

$$dW_i dW_j = \rho dt, \qquad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, N, \tag{2.2}$$

where ρ denotes the correlation coefficient between the i'th and j'th Wiener processes. Since S_t is stochastic, we make use of Itô's lemma in order to derive the process for the contingent claim,

$$dF(t, S_t) = \left(\frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} \right) dt + \frac{\partial F(t, S_t)}{\partial S_t} dS_t.$$

The value of the portfolio at time t is given by

$$\Pi(t) = F(t, S_t) + \Delta(t) S_t,$$

where $\Delta(t) S_t$ denotes the vector inner products, i.e.

$$\Delta(t) S_t = \sum_{i=1}^{N} \Delta_i(t) S_i(t).$$

The dynamics of the portfolio are given by [Björk, 2004]

$$\begin{aligned} d\Pi(t) &= dF(t, S_t) + \Delta(t) dS_t \\ &= \left(\frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} \right) dt + \left(\frac{\partial F(t, S_t)}{\partial S_t} + \Delta(t) \right) dS_t. \end{aligned}$$

By choosing

$$\Delta(t) = \Delta_t = -\frac{\partial F(t, S_t)}{\partial S_t},$$

we obtain an instantaneously risk free portfolio

$$d\Pi(t) = \left(\frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} \right) dt,$$

which earns the risk free rate of interest since it is of the form $dx = ydt$ [Björk, 2004]. We thus have

$$\begin{aligned} r\Pi(t)dt &= \left(\frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} \right) dt \\ \therefore rF(t, S_t) - rS_t \frac{\partial F(t, S_t)}{\partial S_t} &= \frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} \\ \therefore \frac{\partial F(t, S_t)}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 F(t, S_t)}{\partial S_t^2} + rS_t \frac{\partial F(t, S_t)}{\partial S_t} - rF(t, S_t) &= 0. \end{aligned} \tag{2.3}$$

Equation (2.3) is subject to the terminal boundary condition

$$F(T, S_T) = \Phi(S_T),$$

where $\Phi(S_T)$ is the payoff function of the continent claim.

Certain minor variations of the Black and Scholes partial differential equation exist. Hull [2003] derives two variations namely an equation for a claim on a stock that pays a known continuous dividend yield, and an equation for a contingent claim depending on a futures price, or more generally any contingent claim that is continuously margined [Wilmott, 2001]. Hence we adopt a more general version of the Black and Scholes partial differential equation,

$$\begin{aligned} F'_t + \alpha(t, S_t)F''_S + \beta(t, S_t)F'_S + \gamma(t, S_t)F_t + \delta(t, S_t) &= 0 \\ F(T, S_T) &= \Phi(S_T), \end{aligned} \tag{2.4}$$

where

$$\begin{aligned} F'_t &= \frac{\partial F(t, S_t)}{\partial t}, \\ F''_S &= \frac{\partial^2 F(t, S_t)}{\partial S_t}, \\ F'_S &= \frac{\partial F(t, S_t)}{\partial S_t}, \text{ and} \\ F_t &\equiv F(t, S_t), \end{aligned}$$

and $\alpha(t, S_t)$, $\beta(t, S_t)$, $\gamma(t, S_t)$ and $\delta(t, S_t)$ are general functions.

2.3 Modifications of the Black and Scholes partial differential equation

2.3.1 Transformation to an initial value problem

Our first modification of the Black–Scholes partial differential equation is to make the transformation to an initial value problem. We do this by adopting the variable

$$\tau = T - t.$$

Transforming the Black–Scholes partial differential equation to an initial value problem changes equation (2.3) to

$$-F'_\tau + \alpha(\tau, S_\tau)F''_S + \beta(\tau, S_\tau)F'_S + \gamma(\tau, S_\tau)F_\tau + \delta(\tau, S_\tau) = 0, \tag{2.5}$$

with initial boundary condition

$$F(T, S_T) = \Phi(S_T),$$

where $\Phi(S_T)$ is the payoff function of the contingent claim.

2.3.2 Transformation to the heat equation

The convection and diffusion partial differential equation is transformed to a diffusion only heat equation. The reason for the transform is twofold. Firstly the heat equation presents a simpler equation to solve and analyse, and secondly, convection dominant partial differential equations are known to be problematic under certain conditions [see for instance Duffy, 2006b; Seydel, 2004].

We define a new function $V(\zeta, \Xi_\zeta)$ such that

$$F(t, S_t) = e^{a\Xi_\zeta + b\zeta}V(\zeta, \Xi_\zeta),$$

where

$$a = -\frac{1}{2}\left(\frac{2r}{\sigma^2} - 1\right), \quad b = -\frac{1}{4}\left(\frac{2r}{\sigma^2} + 1\right)^2, \quad S = e^{\Xi_\zeta}, \text{ and } t - T - \frac{2\zeta}{\sigma^2},$$

then $V(\zeta, \Xi_\zeta)$ satisfies the basic heat equation [Wilmott, 2000a]

$$\frac{\partial V(\zeta, \Xi_\zeta)}{\partial \zeta} = \frac{\partial^2 V(\zeta, \Xi_\zeta)}{\partial \Xi_\zeta^2}$$
$$V(T, \Xi_T) = \Psi(T, \Xi_T). \tag{2.6}$$

2.4 Conclusion

We derived the Black and Scholes partial differential equation by assuming a portfolio that consists of a contingent claim and its underlying instruments. The resulting partial differential equation is generalized and transformed into an initial value problem and the heat equation.

Chapter 3

A finite differences framework

3.1 Introduction

In this chapter we set out the basic framework for our analysis. We firstly define the discrete environment in which we will conduct testing. Subsequently we proceed to derive difference equations in order to estimate the heat equation (equation 2.6) and the initial value Black and Scholes equation (equation 2.5). In order to simplify our approach we conduct our analysis in a single spatial dimension, i.e.

$$S(\tau) \equiv S_1(\tau), \text{ and}$$
$$\Xi(\zeta) \equiv \Xi_1(\zeta).$$

3.2 The finite difference framework

3.2.1 The initial value Black–Scholes framework

The finite difference grid

We discretise the spatial variable S_τ. Let $s \in [s_\chi, s_\psi]$ be the discretised share price. The interval $[s_\chi, s_\psi]$ is subdivided into $N + 1$ intervals. Each interval is of length

$$h = \frac{s_\psi - s_\chi}{N}$$

Similarly we subdivide the temporal variable τ. Let $q \in [q_\chi = T, q_\psi = t_0]$ be the discretised temporal variable. The interval $q = [q_\chi = T, q_\psi = t_0]$ is subdivided into

$M+1$ intervals. Each interval is of length

$$k = \frac{q_\psi - q_\chi}{M}.$$

subdivisions. We approximate the function $F(\tau, S_\tau)$ with a function $f(q,s)$. We adopt the following notation:

$$f_j^i \equiv f(q_\chi + ik, s_\chi + jh); \qquad i = 0,1,\ldots,N-1,N; \quad j = 0,1,\ldots,M-1,M.$$

We refer to i and j as *mesh* points, and may refer to the (i,j)'th mesh point, meaning the above.

Boundary conditions

The finite difference method utilizes the known values at the boundaries in order to estimate the unknown values. The known values that are provided are the initial condition and the upper and lower boundary conditions. The initial condition is known from the payoff function from equation (2.5), i.e.

$$f(q_\chi, s) = \Phi(s).$$

The upper and lower boundary conditions are found by investigating the properties of the option. The upper and lower boundary conditions may assume any of three categories, namely Dirichlet, Neumann an Robin boundary conditions [Duffy, 2006b]. For an upper boundary s_ψ and lower boundary s_χ, these conditions are given by

$$\begin{aligned}
\text{Dirichlet:} \quad & f(q,s_\psi) = \psi_0, \text{ and} \\
& f(q,s_\chi) = \chi_0, \\
\text{Neumann:} \quad & \tfrac{\partial f(q,s_\psi)}{\partial s} = \psi_0, \text{ and} \\
& \tfrac{\partial f(q,s_\chi)}{\partial s} = \chi_0, \text{ and} \\
\text{Robin:} \quad & x_0 f(q,s_\psi) + x_1 \tfrac{\partial f(q,s_\psi)}{\partial s} = \psi_0, \text{ and} \\
& y_0 f(q,s_\chi) + y_1 \tfrac{\partial f(q,s_\chi)}{\partial s} = \chi_0, \\
& |x_0| + |x_1| \neq 0; |y_0| + |y_1| \neq 0.
\end{aligned}$$

The Robin condition is the most general and both the Dirichlet and Neumann conditions are special cases of the Robin condition.

The domain of $S \in [0,\infty)$ provide us with a convenient lower bound namely $s_\chi = 0$. At this spot price we can determine the function $f(q,s)$ with certainty. The upper

bound is somewhat more problematic. Wilmott [2000b] suggests a Dirichlet condition of "...three or four times the value of the asset at which there is important behavior."

Estimation of the partial derivatives.

The temporal derivative. We estimate the partial derivatives of equation (2.4) by a series of difference equations. From the definition of a derivative we know that for a general function Q on variables x and y,

$$\frac{\partial Q(x,y)}{\partial x} \equiv \lim_{\Delta x \to 0} \frac{Q(x+\Delta x, y) - Q(x,y)}{\Delta x}.$$

We make use of this definition in order to approximate the partial derivative of $f(q,s)$ with respect to time q:

$$\frac{\partial f_j^i}{\partial q} = \lim_{k \to 0} \frac{f_j^{i+k} - f_j^i}{k}.$$

We approximate the partial derivative by assuming that k is *sufficiently* small, i.e.

$$\frac{\partial f_j^i}{\partial q} \approx \frac{f_j^{i+k} - f_j^i}{k}. \tag{3.1}$$

The spatial derivative. We adopt a different approach than the one above to estimate the spatial derivative by making use of a more accurate [Wilmott, 2000b] two sided estimate, i.e.

$$\begin{aligned} \frac{\partial Q(x,y)}{\partial x} &= \lim_{\Delta x \to 0} \frac{1}{2} \left[\frac{Q(x+\Delta x, y) - Q(x,y)}{\Delta x} + \frac{Q(x,y) - Q(x-\Delta x, y)}{\Delta x} \right] \\ &= \lim_{\Delta x \to 0} \frac{Q(x+\Delta x, y) - Q(x-\Delta x, y)}{2\Delta x}. \end{aligned}$$

We use this definition to derive an approximation of the partial derivative of $f(q,s)$ with respect to s,

$$\frac{\partial f_j^i}{\partial s} \approx \frac{f_{j+h}^i - f_{j-h}^i}{2h}. \tag{3.2}$$

The second spatial derivative. We make use of the definition of a second derivative,

$$\begin{aligned} \frac{\partial^2 Q(x,y)}{\partial x^2} &= \lim_{\Delta x \to 0} \frac{\frac{Q(x+\Delta x,y) - Q(x,y)}{\Delta x} - \frac{Q(x,y) - Q(x-\Delta x,y)}{\Delta x}}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{Q(x+\Delta x, y) - 2Q(x,y) + Q(x-\Delta x, y)}{(\Delta x)^2}, \end{aligned}$$

in order to estimate the second derivative of the function $f(q,s)$ with respect to s,

$$\frac{\partial^2 f_j^i}{\partial s^2} \approx \frac{f_{j+h}^i - 2f_j^i + f_{j-h}^i}{h^2}. \tag{3.3}$$

3.2.2 The heat equation framework

The finite difference grid

We approximate the function $V(\zeta, \Xi_\zeta)$ with the function $v(\nu, \xi)$, which is a discrete version of that function. The temporal variable $\nu \in [\nu_\chi, \nu_\psi]$ is subdivided into $M+1$ divisions, each of length

$$\kappa = \frac{\nu_\psi - \nu_\chi}{M}.$$

The spatial variable $\xi \in [\xi_\chi, \xi_\psi]$ is subdivided into $N+1$ divisions, each of length

$$\eta = \frac{\xi_\psi - \xi_\chi}{N}.$$

We adopt the notation

$$v_j^i \equiv v(\nu_\chi + i\kappa, \xi_\chi + j\eta), \qquad i = 0, 1, \ldots, M-1, M; \quad j = 0, 1, \ldots, N-1, N.$$

We refer to i and j as *mesh* points, and may refer to the (i, j)'th mesh point, meaning the above.

Boundary conditions

The initial value is given by equation 2.6, i.e.

$$v(\nu_\chi, \xi) = \Psi(\xi).$$

The upper and lower boundary conditions may again take on any of three forms, i.e.

Dirichlet: $v(\nu, \xi_\psi) = \psi_0$, and
$v(\nu, \xi_\chi) = \chi_0$,

Neumann: $\frac{\partial v(\nu, \xi_\psi)}{\partial s} = \psi_0$, and
$\frac{\partial v(\nu, \xi_\chi)}{\partial s} = \chi_0$, and

Robin: $x_0 v(\nu, \xi_\psi) + x_1 \frac{\partial v(\nu, \xi_\psi)}{\partial \xi} = \psi_0$, and
$y_0 v(\nu, \xi_\chi) + y_1 \frac{\partial v(\nu, \xi_\chi)}{\partial \xi} = \chi_0$,
$|x_0| + |x_1| \neq 0; |y_0| + |y_1| \neq 0.$

Partial derivative approximations

Similar to the initial value Black–Scholes partial differential equation, we make use of a one–sided difference equation to estimate the temporal derivative, and a central

differenc equation in order to estimate the second spatial derivative,

$$\frac{\partial v_j^i}{\partial \nu} \approx \frac{v_j^{i+\kappa} - v_j^i}{\kappa}, \text{ and} \tag{3.4}$$

$$\frac{\partial^2 v_j^i}{\partial \xi^2} \approx \frac{v_{j+\eta}^i - 2v_j^i + v_{j-\eta}^i}{\eta^2}. \tag{3.5}$$

3.3 The explicit finite difference method

3.3.1 The Black–Scholes partial differential equation

We assume the function $F(\tau, S_\tau)$ is the solution to the partial differential equation

$$\begin{aligned} -F'_\tau + \alpha(\tau, S_\tau)F''_S + \beta(\tau, S_\tau)F'_S + \gamma(\tau, S_\tau)F_\tau + \delta(\tau, S_\tau) &= 0 \\ F(T, S_T) &= \Phi(S_T). \end{aligned}$$

We approximate the function $F(\tau, S_\tau)$ with the function $f(q, s)$, which is a solution to the partial differential equation

$$\begin{aligned} -f'_q + \alpha(q,s)f''_s + \beta(q,s)f'_s + \gamma(q,s)f_q + \delta(q,s) &= 0 \\ \text{with } f(q_\chi, s) &= \Phi(s), \\ x_0 f(q, s_\psi) + x_1 \frac{\partial f(q, s_\psi)}{\partial s} &= \psi_0, \text{ and} \\ y_0 f(q, s_\chi) + y_1 \frac{\partial f(q, s_\chi)}{\partial s} &= \chi_0, \end{aligned} \tag{3.6}$$

where

$$\begin{aligned} |x_0| + |x_1| &\neq 0, \text{ and} \\ |y_0| + |y_1| &\neq 0. \end{aligned}$$

Since we have no reasonable way to find the partial derivatives of equation (3.6) we estimate this function by $\hat{f}(q, s)$ which is a solution to the equation

$$\begin{aligned} -\hat{f}'_q + \alpha(q,s)\hat{f}''_s + \beta(q,s)\hat{f}'_s + \gamma(q,s)\hat{f}_q + \delta(q,s) &= 0 \\ \text{with } \hat{f}(q_\chi, s) &= \Phi(s), \\ x_0 \hat{f}(q, s_\psi) + x_1 \frac{\partial \hat{f}(q, s_\psi)}{\partial s} &= \psi_0, \text{ and} \\ y_0 \hat{f}(q, s_\chi) + y_1 \frac{\partial \hat{f}(q, s_\chi)}{\partial s} &= \chi_0. \end{aligned} \tag{3.7}$$

The symbols

$$\hat{f}'_q, \quad \hat{f}''_s, \text{ and} \quad \hat{f}'_s$$

denote estimates to the partial derivatives given in equation (3.6). By substituting these for equations (3.1, 3.3, and 3.2) we obtain

$$-\frac{\hat{f}_j^{i+1}-\hat{f}_j^i}{k}+\alpha(q,s)\frac{\hat{f}_{j+1}^i-2\hat{f}_j^i+\hat{f}_{j-1}^i}{h^2}+\beta(q,s)\frac{\hat{f}_{j+1}^i-\hat{f}_{j-1}^i}{2h}+\gamma(q,s)\hat{f}_j^i+\delta(q,s)=0. \tag{3.8}$$

By rearranging terms, we can explicitly find the value for $\hat{f}_j^{i+1}$ with the difference equation

$$\hat{f}_j^{i+1}=A_j^i\hat{f}_{j+1}^i+(1+B_j^i)\hat{f}_j^i+C_j^i\hat{f}_{j-1}^i+D_j^i, \tag{3.9}$$

where

$$\begin{aligned} A_j^i &= \frac{\alpha(q_\chi+ik,s_\chi+jh)k}{h^2}+\frac{\beta(q_\chi+ik,s_\chi+jh)k}{2h},\\ B_j^i &= \gamma(q_\chi+ik,s_\chi+jh)k-\frac{2\alpha(q_\chi+ik,s_\chi+jh)k}{h^2},\\ C_j^i &= \frac{\alpha(q_\chi+ik,s_\chi+jh)k}{h^2}-\frac{\beta(q_\chi+ik,s_\chi+jh)k}{2h}, \text{ and}\\ D_j^i &= \delta(q_\chi+ik,s_\chi+jh)k. \end{aligned} \tag{3.10}$$

3.3.2 The heat equation

We treat the approximation of the function $V(\zeta,\Xi_\zeta)$ (equation (2.6))in a similar fashion by approximating it with a function $v(\nu,\xi)$. Since the function $v(\nu,\xi)$ requires the exact values of the partial derivatives $v'_\nu(\nu,\xi)$ and $v''_\xi(\nu,\xi)$, we estimate $v(\nu,\xi)$ by a function $\hat{\nu}(\nu,\xi)$ which can be found from the equation

$$\hat{v}'_\nu=\hat{v}''_\xi,$$

where $\hat{v}'_\nu$ and $\hat{v}''_\xi$ are given by equations (3.4) and (3.5) respectively. This yields the equation

$$\frac{\hat{v}_j^{i+1}-\hat{v}_j^i}{\kappa}=\frac{\hat{v}_{j+1}^i-2\hat{v}_j^i+\hat{v}_{j-1}^i}{\eta^2}, \tag{3.11}$$

which, after rearrangement provide a means to find explicitly the value of v_j^{i+1}:

$$\hat{v}_j^{i+1}=\lambda\hat{v}_{j+1}^i+(1-2\lambda)\,\hat{v}_j^i+\lambda\hat{v}_{j-1}^i, \tag{3.12}$$

where

$$\lambda=\frac{\kappa}{\eta^2}.$$

3.4 The implicit finite difference method

Instead of taking a one–sided forward difference estimation for the temporal derivative, we estimate the temporal derivative with a one–side backward equation, i.e.

$$\frac{\partial Q_j^i}{\partial q} \approx \frac{Q_j^i - Q_j^{i-k}}{k}.$$

We proceed by evaluating the derivatives at the temporal step $q_\chi + (i+1)k$ instead of $q_\chi + ik$ in the case of the Black–Scholes equation, and at $\nu_\chi + (i+1)\kappa$ instead of $\nu_\chi + i\kappa$ in the case of the heat equation. This results in a linear set of equations that need to be solved.

3.4.1 The Black–Scholes partial differential equation

The function $\hat{f}(q,s)$ is solved from the equation

$$-\frac{\hat{f}_j^{i+1} - \hat{f}_j^i}{k} + \alpha(q,s)\frac{\hat{f}_{j+1}^{i+1} - 2\hat{f}_j^{i+1} + \hat{f}_{j-1}^{i+1}}{h^2} + \beta(q,s)\frac{\hat{f}_{j+1}^{i+1} - \hat{f}_{j-1}^{i+1}}{2h} + \gamma(q,s)\hat{f}_j^{i+1} + \delta(q,s) = 0. \tag{3.13}$$

By grouping terms we obtain

$$-A_j^{i+1}\hat{f}_{j+1}^{i+1} + (1 - B_j^{i+1})\hat{f}_j^{i+1} - C_j^{i+1}\hat{f}_{j-1}^{i+1} - D_j^{i+1} = \hat{f}_j^i. \tag{3.14}$$

3.4.2 The heat equation

The function $\hat{v}(\nu,\xi)$ is solved from the function

$$\frac{\hat{v}_j^{i+1} - \hat{v}_j^i}{\kappa} = \frac{\hat{v}_{j+1}^{i+1} - 2\hat{v}_j^{i+1} + \hat{v}_{j-1}^{i+1}}{\eta^2}. \tag{3.15}$$

Grouping terms result in

$$-\lambda\hat{v}_{j+1}^{i+1} + (1+2\lambda)\,\hat{v}_j^{i+1} - \lambda_{j-1}^{i+1} = \hat{v}_j^i. \tag{3.16}$$

3.5 The Crank & Nicolson Scheme

3.5.1 The Black–Scholes partial differential equation

Crank and Nicolson [Crank and Nicolson, 1947] inserts a fictitious point on the temporal axis exactly half way between the points (i,j) and $(i+1,j)$. The point $(i+\frac{1}{2},j)$ replaces point $(i+1,j)$ in the explicit scheme and the point (i,j) in the implicit scheme.

A cancelation of the point $(i+\frac{1}{2}, j)$ results in the Crank and Nicolson scheme to be an average of the explicit and implicit schemes. Seydel [2004] adds the explicit and implicit schemes in order to derive the Crank & Nicolson scheme.

We assume $\alpha(q,s) = \alpha$, $\beta(q,s) = \beta$, $\gamma(q,s) = \gamma$ and $\delta(q,s) = \delta = 0$ are constant. Adding equation (3.8) and equation (3.13) we get

$$\begin{aligned} -\frac{\hat{f}_j^{i+1} - \hat{f}_j^i}{k} + \alpha\frac{\hat{f}_{j+1}^{i+1} - 2\hat{f}_j^{i+1} + \hat{f}_{j-1}^{i+1}}{h^2} + \beta\frac{\hat{f}_{j+1}^{i+1} - \hat{f}_{j-1}^{i+1}}{2h} + \gamma\hat{f}_j^{i+1} + \dots \\ -\frac{\hat{f}_j^{i+1} - \hat{f}_j^i}{k} + \alpha\frac{\hat{f}_{j+1}^i - 2\hat{f}_j^i + \hat{f}_{j-1}^i}{h^2} + \beta\frac{\hat{f}_{j+1}^i - \hat{f}_{j-1}^i}{2h} + \gamma\hat{f}_j^i = \\ A\hat{f}_{j+1}^{i+1} + (B-2)\hat{f}_j^{i+1} + C\hat{f}_{j-1}^{i+1} + A\hat{f}_{j+1}^i + (B+2)\hat{f}_j^i + C\hat{f}_{j-1}^i = 0. \end{aligned} \tag{3.17}$$

3.5.2 The heat equation

Adding equation (3.11) and (3.15), results in the Crank & Nicolson difference equation

$$\frac{2}{\kappa}(\hat{v}_j^{i+1} - \hat{v}_j^i) - \frac{1}{\eta^2}(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i + \hat{v}_{j+1}^{i+1} - 2\hat{v}_j^{i+1} + \hat{v}_{j-1}^{i+1}) = 0. \tag{3.18}$$

3.6 A generalized finite difference scheme

Equations (3.9) and (3.14) may be generalized with the equation

$$\begin{aligned} -\theta A_j^{i+1}\hat{f}_{j+1}^{i+1} + (1-\theta B_j^{i+1})\hat{f}_j^{i+1} - \theta C_j^{i+1}\hat{f}_{j-1}^{i+1} - \theta D_j^{i+1} \\ = (1-\theta)A_j^i\hat{f}_{j+1}^i + (1+B_j^i(1-\theta))\hat{f}_j^i + (1-\theta)C_j^i\hat{f}_{j-1}^i + (1-\theta)D_j^i, \end{aligned} \tag{3.19}$$

and similarly equations (3.12) and (3.16) may be generalized with the equation [Shaw]

$$-\theta\lambda\hat{v}_{j+1}^{i+1} + (1+2\lambda\theta)\hat{v}_j^{i+1} - \theta\lambda\hat{v}_{j-1}^{i+1} = (1-\theta)\lambda\hat{v}_{j+1}^i + (1-2\lambda(1-\theta))\hat{v}_j^i + (1-\theta)\hat{v}_{j-1}^i. \tag{3.20}$$

The choice of θ determines whether the scheme is implicit or explicit. Other choices also exist, in particular $\theta = 0.5$ results in the Crank & Nicolson scheme [Crank and Nicolson, 1947].

$$\theta = \begin{cases} 0 & \text{explicit scheme,} \\ \frac{1}{2} & \text{Crank \& Nicolson sheme, and} \\ 1 & \text{implicit scheme.} \end{cases}$$

More efficient choices exist for instance the Douglas scheme, which will be discussed later on.

3.7 Conclusion

We established a framework in which to study the Black and Scholes partial differential equation. This was done by discretising the function on a grid with spatial and temporal axes. Estimates for the partial derivatives were found and are listed in Table 3.1 These were substituted into the discretised version of the partial differential equation such that the discrete version of the partial differential equation can be represented by a series of difference equations. Three finite difference schemes were derived namely the fully explicit scheme, the fully implicit scheme and the Crank and Nicolson scheme. We finally derived a generalization of the three schemes.

Derivative	**Black–Scholes PDE**	**Heat equation**
Temporal derivative	$\frac{f_j^{i+k}-f_j^i}{k}$	$\frac{v_j^{i+\kappa}-v_j^i}{\kappa}$
Second order spatial derivative	$\frac{f_{j+h}^i-2f_j^i+f_{j-h}^i}{h^2}$	$\frac{v_{j+\eta}^i-2v_j^i+v_{j-\eta}^i}{\eta^2}$
Spatial derivative	$\frac{f_{j+h}^i-f_{j-h}^i}{2h}$	N/A

Table 3.1: A summary of estimations of derivatives.

Chapter 4

Truncation error, consistency and stability

4.1 Introduction

In this chapter we investigate the important properties of truncation error, consistency, convergence and stability.

4.2 Local truncation error

The local truncation error measures by how much the approximating difference equation does not satisfy the original partial differential equation at the mesh points i and j. Smith [1984] provides the following treatment:
Let $G_j^i(\hat{f}) = 0$ represent the difference equation approximating the partial differential equation at the (i, j)'th mesh point. If we replace $\hat{f}$ by F, where F is the exact solution of the partial differential equation, the value $G_j^i(F)$ is called the local truncation error,

$$T_j^i = G_j^i(F). \tag{4.1}$$

Using Taylor expansions, it is easy to express T_j^i in terms of powers of k and h.

4.2.1 Local truncation error for the initial value Black and Scholes schemes.

We calculate the local truncation error for the partial differential equation

$$F'_t + \alpha(t, S_t)F''_S + \beta(t, S_t)F'_S + \gamma(t, S_t)F_t + \delta(t, S_t) = 0$$

at the mesh point (i, j) for the three classical schemes namely the explicit, implicit and Crank & Nicolson schemes.

The explicit scheme

$$\begin{aligned} G^i_j(\hat{f}) &= -\frac{\hat{f}^{i+1}_j - \hat{f}^i_j}{k} + \alpha(q,s)\frac{\hat{f}^i_{j+1} - 2\hat{f}^i_j + \hat{f}^i_{j-1}}{h^2} \dots \\ &\quad \dots + \beta(q,s)\frac{\hat{f}^i_{j+1} - \hat{f}^i_{j-1}}{2h} + \gamma(q,s)\hat{f}^i_j + \delta(q,s) \end{aligned} \tag{4.2}$$

Substituting F for $\hat{f}$ we obtain

$$\begin{aligned} T^i_j = G^i_j(F) &= -\frac{F^{i+1}_j - F^i_j}{k} + \alpha(q,s)\frac{F^i_{j+1} - 2F^i_j + F^i_{j-1}}{h^2} + \dots \\ &\quad \dots \beta(q,s)\frac{F^i_{j+1} - F^i_{j-1}}{2h} + \gamma(q,s)F^i_j + \delta(q,s) \end{aligned} \tag{4.3}$$

By Taylor's expansion we have the following:

$$\begin{aligned} T^i_j &= -\frac{1}{k}(F^i_j + k\frac{\partial F^i_j}{\partial q} + \frac{1}{2!}k^2\frac{\partial^2 F^i_j}{\partial q^2} + \frac{1}{3!}k^3\frac{\partial^3 F^i_j}{\partial q^3} + \dots - F^i_j) + \dots \\ &\quad \frac{\alpha(q,s)}{h^2}(F^i_j + h\frac{\partial F^i_j}{\partial s} + \frac{1}{2!}h^2\frac{\partial^2 F^i_j}{\partial s^2} + \frac{1}{3!}h^3\frac{\partial^3 F^i_j}{\partial s^3} + \frac{1}{4!}h^4\frac{\partial^4 F^i_j}{\partial s^4} + \dots \\ &\quad -2F^i_j + F^i_j - h\frac{\partial F^i_j}{\partial s} + \frac{1}{2!}h^2\frac{\partial^2 F^i_j}{\partial s^2} - \frac{1}{3!}h^3\frac{\partial^3 F^i_j}{\partial s^3} + \frac{1}{4!}h^4\frac{\partial^4 F^i_j}{\partial s^4} + \dots) \\ &\quad +\frac{\beta(q,s)}{2h}(F^i_j + h\frac{\partial F^i_j}{\partial s} + \frac{1}{2!}h^2\frac{\partial^2 F^i_j}{\partial s^2} + \frac{1}{3!}h^3\frac{\partial^3 F^i_j}{\partial s^3} + \frac{1}{4!}h^4\frac{\partial^4 F^i_j}{\partial s^4} + \dots \\ &\quad -F^i_j + h\frac{\partial F^i_j}{\partial s} - \frac{1}{2!}h^2\frac{\partial^2 F^i_j}{\partial s^2} + \frac{1}{3!}h^3\frac{\partial^3 F^i_j}{\partial s^3} - \frac{1}{4!}h^4\frac{\partial^4 F^i_j}{\partial s^4} + \dots) \\ &\quad +\gamma(q,s)F^i_j + \delta(q,s) \\ &= \left(-\frac{\partial F^i_j}{\partial q} + \alpha(q,s)\frac{\partial^2 F^i_j}{\partial s^2} + \beta(q,s)\frac{\partial F^i_j}{\partial s} + \gamma(q,s)F^i_j + \delta(q,s)\right) + \dots \\ &\quad -\frac{1}{2}k\frac{\partial^2 F^i_j}{\partial q^2} - \frac{1}{6}k^2\frac{\partial^3 F^i_j}{\partial q^3} + \dots + \frac{\alpha(q,s)}{12}h^2\frac{\partial^4 F^i_j}{\partial s^4} + \frac{\alpha(q,s)}{360}h^4\frac{\partial^6 F^i_j}{\partial s^6} + \dots \\ &\quad \dots + \frac{\beta(q,s)}{6}h^2\frac{\partial^3 F^i_j}{\partial s^3} + \frac{\beta(q,s)}{120}h^4\frac{\partial^5 F^i_j}{\partial s^5} + \dots \end{aligned}$$

Since F is the solution to the partial differential equation we have

$$-\frac{\partial F_j^i}{\partial q} + \alpha(q,s)\frac{\partial^2 F_j^i}{\partial s^2} + \beta(q,s)\frac{\partial F_j^i}{\partial s} + \gamma(q,s)F_j^i + \delta(q,s) = 0.$$

The principal part of the truncation error is thus

$$T_j^i = -\frac{1}{2}k\frac{\partial^2 F_j^i}{\partial q^2} + \frac{1}{12}h^2\left(2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\partial^4 F_j^i}{\partial s^4}\right).$$

Hence

$$T_j^i = \mathcal{O}(k) + \mathcal{O}(h^2).$$

The implicit scheme

Similar to the analysis above, the truncation error of the implicit scheme is given by

$$\begin{aligned} T_j^i \quad = \quad & \left(\frac{\partial F_j^i}{\partial q} + \alpha(q,s)\frac{\partial^2 F_j^i}{\partial s^2} + \beta(q,s)\frac{\partial F_j^i}{\partial s} + \gamma(q,s)F_j^i + \delta(q,s)\right) + \dots \\ & -\frac{1}{2}k\frac{\partial^2 F_j^i}{\partial q^2} - \frac{1}{6}k^2\frac{\partial^3 F_j^i}{\partial q^3} + \dots + \frac{\alpha(q,s)}{12}h^2\frac{\partial^4 F_j^i}{\partial s^4} + \frac{\alpha(q,s)}{360}h^4\frac{\partial^6 F_j^i}{\partial s^6} + \dots \\ & \dots + \frac{\beta(q,s)}{6}h^2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\beta(q,s)}{120}h^4\frac{\partial^5 F_j^i}{\partial s^5} + \dots \end{aligned}$$

This has a principal error of

$$T_j^i = -\frac{1}{2}k\frac{\partial^2 F_j^i}{\partial q^2} + \frac{1}{12}h^2\left(2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\partial^4 F_j^i}{\partial s^4}\right).$$

Hence

$$T_j^i = \mathcal{O}(k) + \mathcal{O}(h^2).$$

Crank & Nicolson scheme

In order to ease readability, we omit some super– and subscripts.

$$\begin{aligned} G_j^i(\hat{f}) \quad = \quad & -\frac{1}{k}(\hat{f}_j^{i+1} - \hat{f}_j^i) + \frac{\alpha}{h^2}(\hat{f}_{j+1}^{i+1} - 2\hat{f}_j^{i+1} + \hat{f}_{j-1}^{i+1}) + +\frac{\beta}{2h}(\hat{f}_{j+1}^{i+1} - \hat{f}_{j-1}^{i+1}) + \gamma\hat{f}_j^{i+1} + \dots \\ & -\frac{1}{k}(\hat{f}_j^i - \hat{f}_j^i) + \frac{\alpha}{h^2}(\hat{f}_{j+1}^i - 2\hat{f}_j^i + \hat{f}_{j-1}^i) + +\frac{\beta}{2h}(\hat{f}_{j+1}^i - \hat{f}_{j-1}^i) + \gamma\hat{f}_j^i \end{aligned}$$

Substituting F for $\hat{f}$ we obtain

$$\begin{aligned} T_j^i = G_j^i(F) \quad = \quad & -\frac{1}{k}(F_j^{i+1} - F_j^i) + \frac{\alpha}{h^2}(F_{j+1}^{i+1} - 2F_j^{i+1} + F_{j-1}^{i+1}) + +\frac{\beta}{2h}(F_{j+1}^{i+1} - F_{j-1}^{i+1}) + \gamma F_j^{i+1} + \dots \\ & -\frac{1}{k}(F_j^i - F_j^i) + \frac{\alpha}{h^2}(F_{j+1}^i - 2F_j^i + F_{j-1}^i) + +\frac{\beta}{2h}(F_{j+1}^i - F_{j-1}^i) + \gamma F_j^i \end{aligned} \tag{4.4}$$

After some algebraic manipulation we arrive at

$$\begin{aligned} T_j^i &= 2\left(-\frac{\partial F}{\partial q}+\alpha\frac{\partial^2}{\partial s^2}+\beta\frac{\partial F}{\partial S}+\gamma F\right)+k\left(-\frac{\partial^2 F}{\partial q^2}+\alpha\frac{\partial^3 F}{\partial q\partial s^2}+\beta\frac{\partial^2 F}{\partial q\partial s}+\gamma\frac{\partial F}{\partial q}\right) \\ &+k^2\left(\frac{1}{2}\gamma\frac{\partial^2 F}{\partial q^2}-\frac{1}{3}\frac{\partial^3 F}{\partial q^3}+\frac{1}{2}\gamma\frac{\partial^3 F}{\partial q^2\partial s}+\frac{1}{2}\alpha\frac{\partial^4 F}{\partial q^2\partial s^2}\right) \\ &+h^2\left(\frac{1}{3}\beta\frac{\partial^3 F}{\partial s^3}+\frac{1}{6}\alpha\frac{\partial^4 F}{\partial s^4}\right) \\ &+k^3\left(\frac{1}{6}\gamma\frac{\partial^3 F}{\partial q^3}-\frac{1}{12}\frac{\partial^4 F}{\partial q^4}+\frac{1}{6}\beta\frac{\partial^4 F}{\partial q^3\partial s}+\frac{1}{6}\alpha\frac{\partial^5 F}{\partial q^3\partial s^2}\right) \\ &+kh^2\left(\frac{1}{12}\alpha\frac{\partial^5 F}{\partial q\partial s^4}+\frac{1}{6}\beta\frac{\partial^4 F}{\partial q\partial s^3}\right)+\ldots. \end{aligned}$$

We note

$$k\left(-\frac{\partial^2 F}{\partial q^2}+\alpha\frac{\partial^3 F}{\partial q\partial s^2}+\beta\frac{\partial^2 F}{\partial q\partial s}+\gamma F\right)=k\frac{\partial}{\partial q}\left(-\frac{\partial F}{\partial q}+\alpha\frac{\partial^2}{\partial s^2}+\beta\frac{\partial F}{\partial S}+\gamma F\right),$$

and since F is the solution to the partial differential equation we have

$$\begin{aligned} 2\left(-\frac{\partial F}{\partial q}+\alpha\frac{\partial^2}{\partial s^2}+\beta\frac{\partial F}{\partial S}+\gamma F\right)&=0, \text{ and} \\ k\left(-\frac{\partial^2 F}{\partial q^2}+\alpha\frac{\partial^3 F}{\partial q\partial s^2}+\beta\frac{\partial^2 F}{\partial q\partial s}+\gamma\frac{\partial F}{\partial q}\right)&=0. \end{aligned}$$

The principal part of the truncation error can thus be summarised as

$$T_j^i=\mathcal{O}(k^2)+\mathcal{O}(h^2).$$

4.2.2 Local truncation error for the heat equation schemes.

We calculate the local truncation error for the partial differential equation

$$V'_\nu-F''_\xi=0$$

at the mesh point (i,j) for the three classical schemes namely the explicit, implicit and Crank & Nicolson schemes.

The explicit scheme

The local truncation error for the explicit scheme follows from

$$\begin{aligned} G_j^i(\hat{v}) &= \frac{1}{\kappa}(\hat{v}_j^{i+1} - \hat{v}_j^i) - \frac{1}{\eta^2}(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i) \\ T_j^i = G_j^i(V) &= \frac{1}{\kappa}(V_j^{i+1} - V_j^i) - \frac{1}{\eta^2}(V_{j+1}^i - 2V_j^i + V_{j-1}^i) \\ &= \frac{1}{\kappa}\left(\kappa\frac{\partial V}{\partial \nu} + \frac{1}{2}\kappa^2\frac{\partial^2 V}{\partial \nu^2} + \ldots\right)\ldots \\ &\quad + \frac{1}{\eta^2}(\eta\frac{\partial V}{\partial \xi} + \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial \xi^2} + \frac{1}{6}\eta^3\frac{\partial^3 V}{\partial \xi^3} + \frac{1}{24}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots\ldots \\ &\quad - \eta\frac{\partial V}{\partial \xi} + \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial \xi^2} - \frac{1}{6}\eta^3\frac{\partial^3 V}{\partial \xi^3} + \frac{1}{24}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots) \\ &= \frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2} + \frac{1}{2}\kappa\frac{\partial^2 V}{\partial \nu^2} + \ldots + \frac{1}{12}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots. \end{aligned} \tag{4.5}$$

Since V is the solution to $\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}$ the principal truncation error is summarised as

$$T_j^i = \mathcal{O}(\kappa) + \mathcal{O}(\eta^2).$$

The implicit scheme

The local truncation error for the implicit scheme is derived in similar fashion than for the explicit scheme.

$$\begin{aligned} G_j^i(\hat{v}) &= \frac{1}{\kappa}(\hat{v}_j^i - \hat{v}_j^{i-1}) - \frac{1}{\eta^2}(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i) \\ T_j^i = G_j^i(V) &= \frac{1}{\kappa}(V_j^i - V_j^{i-1}) - \frac{1}{\eta^2}(V_{j+1}^i - 2V_j^i + V_{j-1}^i) \\ &= \frac{1}{\kappa}\left(\kappa\frac{\partial V}{\partial \nu} - \frac{1}{2}\kappa^2\frac{\partial^2 V}{\partial \nu^2} + \ldots\right)\ldots \\ &\quad + \frac{1}{\eta^2}(\eta\frac{\partial V}{\partial \xi} + \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial \xi^2} + \frac{1}{6}\eta^3\frac{\partial^3 V}{\partial \xi^3} + \frac{1}{24}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots\ldots \\ &\quad - \eta\frac{\partial V}{\partial \xi} + \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial \xi^2} - \frac{1}{6}\eta^3\frac{\partial^3 V}{\partial \xi^3} + \frac{1}{24}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots) \\ &= \frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2} - \frac{1}{2}\kappa\frac{\partial^2 V}{\partial \nu^2} + \ldots + \frac{1}{12}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots. \end{aligned} \tag{4.6}$$

Since V is the solution to $\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}$ the principal truncation error is summarised as

$$T_j^i = \mathcal{O}(\kappa) + \mathcal{O}(\eta^2).$$

The Crank & Nicolson scheme

$$\begin{aligned} G_j^i(\hat{v}) &= \frac{2}{\kappa}(\hat{v}_j^{i+1} - \hat{v}_j^i) - \frac{1}{\eta^2}(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i + \hat{v}_{j+1}^{i+1} - 2\hat{v}_j^{i+1} + \hat{v}_{j-1}^{i+1}) \\ T_j^i = G_j^i(V) &= \frac{2}{\kappa}(V_j^{i+1} - V_j^i) - \frac{1}{\eta^2}(V_{j+1}^i - 2V_j^i + V_{j-1}^i + V_{j+1}^{i+1} - 2V_j^{i+1} + V_{j-1}^{i+1}) \\ &= \frac{2}{\kappa}\left(\kappa\frac{\partial V}{\partial \nu} + \frac{1}{2}\kappa^2\frac{\partial^2 V}{\partial \nu^2} + \frac{1}{6}\kappa^3\frac{\partial^3 V}{\partial \nu^3} + \ldots\right)\ldots \\ &\quad -\frac{1}{\eta^2}(2\eta^2\frac{\partial^2 V}{\partial \xi^2} + \kappa\eta^2\frac{\partial^3 V}{\partial v \partial \xi^2} + \frac{1}{2}\kappa^2\eta^2\frac{\partial^4 V}{\partial \nu^2 \partial \xi^2} + \frac{1}{6}\eta^4\frac{\partial^4 V}{\partial \xi^4} + \ldots) \\ &= 2\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) + \kappa\left(\frac{\partial^2 V}{\partial \nu^2} + \frac{\partial^3 V}{\partial \nu \partial \xi^2}\right)\ldots \\ &\quad +\kappa^2\left(\frac{1}{3}\frac{\partial^3 V}{\partial \nu^3} - \frac{1}{2}\frac{\partial^4 V}{\partial \nu^\partial \xi^2}\right) + \eta^2\left(\frac{1}{6}\frac{\partial^4 V}{\partial \xi^4}\right) + \ldots. \end{aligned} \tag{4.7}$$

Since V is the solution to $\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}$, the local truncation error is of the order

$$T_j^i = \mathcal{O}(\kappa^2) + \mathcal{O}(\eta^2).$$

4.3 Douglas schemes

In section 3.6 we compressed the explicit, implicit and Crank & Nicolson schemes into a general scheme, where the parameter θ determines the applicable scheme. In this section we derive a fourth scheme which minimizes the truncation error.

The heat equation

We rewrite equation (3.20) as

$$G_j^i(\hat{v}) = \frac{1}{\eta^2}(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i) - \frac{1}{\eta^2}\theta(\hat{v}_{j+1}^i - 2\hat{v}_j^i + \hat{v}_{j-1}^i) + \frac{1}{\kappa}(\hat{v}_j^i - \hat{v}_j^{i+1}) + \frac{1}{\eta^2}\theta(\hat{v}_{j+1}^{i+1} + \hat{v}_j^{i+1} + \hat{v}_{j-1}^{i+1}).$$

The local truncation error is then

$$\begin{aligned}
T_j^i = G_j^i(V) &= \frac{1}{\eta^2}(V_{j+1}^i - 2V_j^i + V_{j-1}^i) - \frac{1}{\eta^2}\theta(V_{j+1}^i - 2V_j^i + V_{j-1}^i)\dots \\
&\quad + \frac{1}{\kappa}(V_j^i - V_j^{i+1}) + \frac{1}{\eta^2}\theta(V_{j+1}^{i+1} + V_j^{i+1} + V_{j-1}^{i+1}) \\
&= \left(\frac{\partial^2 V}{\partial \xi^2} + \frac{1}{12}\eta^2\frac{\partial^4 V}{\partial \xi^4} + \frac{1}{360}\eta^4\frac{\partial^6 V}{\partial \xi^6} + \dots\right) + \left(-\frac{\partial V}{\partial \nu} - \frac{1}{2}\kappa\frac{\partial^V}{\partial \nu^2} - \frac{1}{6}\kappa^2\frac{\partial^3 V}{\partial \nu^3} + \dots\right)\dots \\
&\quad + \theta\left(-\frac{\partial^2 V}{\partial \xi^2} - \frac{1}{12}\eta^2\frac{\partial^4 V}{\partial \xi^4} - \frac{1}{360}\eta^4\frac{\partial^6 V}{\partial \xi^6} + \dots\right)\dots \\
&\quad + \theta\left(\frac{\partial^2 V}{\partial \xi^2} + \kappa\frac{\partial^3 V}{\partial \nu \partial \xi^2} + \frac{1}{12}\eta^2\frac{\partial^4 V}{\partial \xi^4} + \frac{1}{6}\kappa^3\frac{\partial^5 V}{\partial \nu^3 \partial \xi^2} + \frac{1}{12}\kappa\eta^2\frac{\partial^5 V}{\partial \nu \partial \xi^4} + \dots\right) \\
T_j^i &= -\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) - \kappa\left(\frac{1}{2}\frac{\partial^2 V}{\partial \nu^2} - \theta\frac{\partial^3 V}{\partial \nu \partial \xi^2}\right) + \frac{1}{12}\eta^2\frac{\partial^4 V}{\partial \xi^4} - \frac{1}{6}\kappa^2\frac{\partial^3 V}{\partial \nu^3}\dots \\
&\quad + \left(\frac{1}{2}\kappa^2\theta\frac{\partial^4 V}{\partial \nu^2 \partial \xi^2} + \frac{1}{12}\theta\kappa\eta 2\frac{\partial^5 V}{\partial \nu \partial \xi^4}\right) + \kappa^3\left(-\frac{1}{24}\frac{\partial^4 V}{\partial \nu^4} + \frac{1}{6}\theta\frac{\partial^5 V}{\partial \nu^3 \partial \xi^5}\right) + \dots \qquad (4.8)
\end{aligned}$$

A choice for θ of

$$\theta = \frac{1}{2} - \frac{1}{12}\frac{\eta^2}{\kappa},$$

results in a truncation error of

$$\begin{aligned}
T_j^i &= -\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) - \frac{1}{2}\kappa\frac{\partial}{\partial \nu}\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) - \frac{1}{12}\eta^2\frac{\partial^2}{\partial \xi^2}\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) - \frac{1}{6}\kappa^2\frac{\partial^3 V}{\partial \nu^3} \\
&\quad - \frac{1}{24}\kappa\eta^2\frac{\partial^3}{\partial \nu \partial \xi^2}\left(\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}\right) + \kappa^2\left(\frac{\partial^4 V}{\partial \nu^2 \partial \xi^2}\right) - \eta^4\left(\frac{1}{144}\frac{\partial^5 V}{\partial \nu \partial \xi^4}\right) + \dots.
\end{aligned}$$

Since V is the solution to the equation $\frac{\partial V}{\partial \nu} - \frac{\partial^2 V}{\partial \xi^2}$, the local truncation error is summarised by

$$T_j^i = \mathcal{O}(\kappa^2) + \mathcal{O}(\eta^4).$$

The initial value Black and Scholes equation

The local truncation error for the generalized initial value Black and Scholes equation is given as

$$\begin{aligned}
T_j^i &= k\left(\alpha\theta\frac{\partial^3 F}{\partial q \partial s^2} + \beta\theta\frac{\partial^2 F}{\partial q \partial s} - \frac{1}{2}\frac{\partial^2 F}{\partial q^2} + \gamma\theta\frac{\partial F}{\partial q}\right)\dots \\
&\quad + k^2\left(\frac{1}{2}\alpha\theta\frac{\partial^4 F}{\partial q^2 \partial s^2} + \frac{1}{2}\beta\theta\frac{\partial^3 F}{\partial q^2 \partial s} - \frac{1}{6}\frac{\partial^3 F}{\partial q^3} + \frac{1}{2}\gamma\theta\frac{\partial^2 F}{\partial q^2}\right)\dots \\
&\quad + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3}{\partial s^3}\right)\dots \\
&\quad + kh^2\left(\frac{1}{12}\alpha\theta\frac{\partial^4 F}{\partial q \partial s^4} + \frac{1}{6}\beta\theta\frac{\partial^4}{\partial q \partial s^3}\right)\dots \\
&\quad + h^4\left(\frac{1}{360}\alpha\frac{\partial^6 F}{\partial s^6} + \frac{1}{120}\beta\frac{\partial^5}{\partial s^5}\right) + \dots \qquad (4.9)
\end{aligned}$$

We were not able to find a value for θ that improves on the $\mathcal{O}(k^2) + \mathcal{O}(h^2)$ local truncation error of the Crank & Nicolson scheme. The reason provided by Smith [1984] is that only second-order derivatives allows for the elimination of the fourth order differences. First order derivatives' accuracy can only be improved by involving additional grid points, which complicates boundary conditions for implicit schemes.

4.4 Consistency

A scheme is considered to be consistent with the partial differential equation if the truncation error tend to 0 when the time and spatial steps tend to zero [Smith, 1984].

4.4.1 The initial value Black and Scholes equation schemes

We introduce the variable z such that

$$k = zh.$$

The truncation error of the generalized scheme is

$$\begin{aligned} T_j^i &= zh\left(\alpha\theta\frac{\partial^3 F}{\partial q\partial s^2} + \beta\theta\frac{\partial^2 F}{\partial q\partial s} - \frac{1}{2}\frac{\partial^2 F}{\partial q^2} + \gamma\theta\frac{\partial F}{\partial q}\right)\ldots + \\ &\quad + z^2h^2\left(\frac{1}{2}\alpha\theta\frac{\partial^4 F}{\partial q^2\partial s^2} + \frac{1}{2}\beta\theta\frac{\partial^3 F}{\partial q^2\partial s} - \frac{1}{6}\frac{\partial^3 F}{\partial q^3} + \frac{1}{2}\gamma\theta\frac{\partial^2 F}{\partial q^2}\right)\ldots + \\ &\quad + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3}{\partial s^3}\right)\ldots + \\ &\quad + zh^3\left(\frac{1}{12}\alpha\theta\frac{\partial^4 F}{\partial q\partial s^4} + \frac{1}{6}\beta\theta\frac{\partial^4}{\partial q\partial s^3}\right)\ldots + \\ &\quad + h^4\left(\frac{1}{360}\alpha\frac{\partial^6 F}{\partial s^6} + \frac{1}{120}\beta\frac{\partial^5}{\partial s^5}\right) + \ldots \end{aligned}$$

Clearly

$$\lim_{h\to 0} T_j^i = 0,$$

irrespective of the choice of θ. The implicit, explicit, Crank and Nicolson and Douglas schemes are all consistent with the partial differential equation.

4.4.2 The heat equation schemes

We introduce the variable ω such that

$$\kappa = \omega\eta.$$

The local truncation error for the generalized scheme for the heat equation is

$$T_j^i = -\frac{1}{6}\omega^2\eta^2\frac{\partial^3 V}{\partial\nu^3} + \omega^2\eta^2\left(\frac{\partial^4 V}{\partial\nu^2\partial\xi^2}\right) - \eta^4\left(\frac{1}{144}\frac{\partial^5 V}{\partial\nu\partial\xi^4}\right) + \ldots.$$

Since

$$\lim_{\eta\to 0} T_j^i = 0,$$

we conclude that the generalized scheme is consistent with the heat equation for all values of θ.

4.5 Stability

Tavella and Randall [2000] provides the following description for stability:

> A numerical scheme is said to be stable if the difference between the numerical solution and the exact solution remains bounded as the number to time steps tend to infinity.

Stability is a computational problem. Computers have limited capacity to store numbers with no concept of real numbers, subsequently small rounding errors result when difference equations are computed.

As long as these errors remain bounded from one temporal step to the next, the scheme is stable. However, if the rounding errors perpetuate and grow with each temporal step, the scheme may become unstable returning values with no practical use.
We consider two methods to determine whether a scheme is stable. The first follows matrix analysis discussed by Smith [1984], Seydel [2004] and Geske and Shastri [1985], amongst others, and was originally derived by Richtmeyer and Lax [Smith, 1984]. The second analysis is based on Fourier analysis and is discussed amongst others by Wilmott [2000b]; Smith [1984]. It also closely follows the argument of Du Fort and Frankel [1953] presented in Chapter 5.

4.5.1 Matrix method to determine stability

We write the general equation (3.19) as

$$\begin{aligned}
\mathbf{M_L f^{i+1}} + \mathbf{b_L^{i+1}} &= \mathbf{M_R f^i} + \mathbf{b_R^i} \\
\therefore \mathbf{f^{i+1}} &= \left(\mathbf{M_R f^i} + \mathbf{b_R^i} - \mathbf{b_L^{i+1}}\right)\mathbf{M_L^{-1}} \\
&= \mathbf{M f^i} + \tilde{\mathbf{b}}^{\mathbf{i}}, \qquad (4.10)
\end{aligned}$$

where

$$\mathbf{M_L} = \begin{pmatrix} 1-\theta B & -\theta A & & \\ -\theta C & & 0 & \\ & \ddots & & \\ & 0 & & -\theta A \\ & & -\theta C & 1-\theta B \end{pmatrix},$$

$$\mathbf{M_R} = \begin{pmatrix} 1+(1-\theta)B & (1-\theta)A & & \\ (1-\theta)C & & 0 & \\ & \ddots & & \\ & 0 & & (1-\theta)A \\ & & (1-\theta)C & 1+(1-\theta)B \end{pmatrix},$$

$$\mathbf{f^x} = \begin{pmatrix} \hat{f}_1^x \\ \hat{f}_2^x \\ \vdots \\ \hat{f}_{N-2}^x \\ \hat{f}_{N-1}^x \end{pmatrix}, \quad \mathbf{b_L^x} = \begin{pmatrix} -\theta C \hat{f}_0^x \\ 0 \\ \vdots \\ 0 \\ -\theta A \hat{f}_N^x \end{pmatrix}, \quad \mathbf{b_R^x} = \begin{pmatrix} (1-\theta) C \hat{f}_0^x \\ 0 \\ \vdots \\ 0 \\ (1-\theta) A \hat{f}_N^x \end{pmatrix},$$

and where $\mathbf{M_L^{-1}}$ denotes the inverse of matrix $\mathbf{M_L}$, $\mathbf{M} \equiv \mathbf{M_L^{-1}M_R}$, and $\tilde{\mathbf{b}}^{\mathbf{i}} \equiv \mathbf{M_L^{-1}}(\mathbf{b_R^i} - \mathbf{b_L^{i+1}})$.

Similarly we write the generalized equation (3.20) as

$$\mathbf{\Sigma_L v^{i+1}} + \beta_{\mathbf{L}}^{\mathbf{i+1}} \quad = \quad \mathbf{\Sigma_R v^i} + \beta_{\mathbf{R}}^{\mathbf{i}}.$$

Rearranging terms give

$$\begin{aligned} \mathbf{v^{i+1}} &= \left(\mathbf{\Sigma_R v^i} + \beta_{\mathbf{R}}^{\mathbf{i}} - \beta_{\mathbf{L}}^{\mathbf{i+1}}\right) \mathbf{\Sigma_L^{-1}} \\ &= \mathbf{\Sigma v^i} + \tilde{\beta}^{\mathbf{i}}, \end{aligned} \tag{4.11}$$

where

$$\mathbf{\Sigma_L} = \begin{pmatrix} 1+2\theta\lambda & -\theta\lambda & & & \\ -\theta\lambda & & & 0 & \\ & & \ddots & & \\ & & 0 & & -\theta\lambda \\ & & & -\theta\lambda & 1+2\theta\lambda \end{pmatrix},$$

$$\mathbf{\Sigma_R} = \begin{pmatrix} 1-2(1-\theta)\lambda & (1-\theta)\lambda & & & \\ (1-\theta)\lambda & & & 0 & \\ & & \ddots & & \\ & & 0 & & (1-\theta)\lambda \\ & & & (1-\theta)\lambda & 1-2(1-\theta)\lambda \end{pmatrix},$$

$$\mathbf{v^x} = \begin{pmatrix} \hat{v}_1^x \\ \hat{v}_2^x \\ \vdots \\ \hat{v}_{N-2}^x \\ \hat{v}_{N-1}^x \end{pmatrix}, \quad \beta_{\mathbf{L}}^{\mathbf{x}} = \begin{pmatrix} -\theta\lambda\hat{v}_0^x \\ 0 \\ \vdots \\ 0 \\ -\theta\lambda\hat{v}_N^x \end{pmatrix}, \quad \beta_{\mathbf{R}}^{\mathbf{x}} = \begin{pmatrix} (1-\theta)\lambda\hat{v}_0^x \\ 0 \\ \vdots \\ 0 \\ (1-\theta)\lambda\hat{v}_N^x \end{pmatrix},$$

and where $\mathbf{\Sigma_L^{-1}}$ denotes the inverse of matrix $\mathbf{\Sigma_L}$, $\mathbf{\Sigma} \equiv \mathbf{\Sigma_L^{-1}\Sigma_R}$, and $\tilde{\beta}^{\mathbf{i}} \equiv \mathbf{\Sigma_L^{-1}}(\beta_{\mathbf{R}}^{\mathbf{i}} - \beta_{\mathbf{L}}^{\mathbf{i+1}})$.

Recursively equation 4.10 may be written as

$$\begin{aligned} \mathbf{f^{i+1}} &= \mathbf{Mf^i} + \tilde{\mathbf{b^i}} \\ &= \mathbf{M}(\mathbf{Mf^{i-1}} + \tilde{\mathbf{b^{i-1}}}) + \tilde{\mathbf{b^i}} \\ &= \mathbf{M^2 f^{i-1}} + \mathbf{M}\tilde{\mathbf{b^{i-1}}} + \tilde{\mathbf{b^i}} \\ &= \dots \\ &= \mathbf{M}^{i+1}\mathbf{f^0} + \mathbf{M}^i\tilde{\mathbf{b^0}} + \mathbf{M}^{i-1}\tilde{\mathbf{b^1}} + \dots + \tilde{\mathbf{b^i}}, \end{aligned} \tag{4.12}$$

where $\mathbf{f^0}$ is the vector of initial boundary values, and $\tilde{\mathbf{b^y}}, \quad y = 0, 1, \dots, i$ are the vectors of known upper and lower boundary values. Let $\tilde{\mathbf{f^i}}$ be the vector of computed values for $\hat{f}^i$. There rounding error is then

$$\mathbf{e^i} = \mathbf{f^i} - \tilde{\mathbf{f}}^{\mathbf{i}}.$$

If we assume that a perturbation occurred in the vector of initial values, i.e. $\tilde{\mathbf{f}}^{\mathbf{0}} =$

$\mathbf{f^0} + \mathbf{e^0}$, then equation (4.12) may be written as

$$\tilde{\mathbf{f}}^{\mathbf{i+1}} = \mathbf{M}^{i+1}\tilde{\mathbf{f}}^{\mathbf{0}} + \mathbf{M}^{i}\tilde{\mathbf{b}}^{\mathbf{0}} + \mathbf{M}^{i-1}\tilde{\mathbf{b}}^{\mathbf{1}} + \ldots + \tilde{\mathbf{b}}^{\mathbf{i}}. \tag{4.13}$$

Subtracting equation (4.13) from equation (4.12) results in

$$\mathbf{e}^{\mathbf{i+1}} = \mathbf{M}^{i+1}(\mathbf{f}^{\mathbf{0}} - \tilde{\mathbf{f}}^{\mathbf{0}}) = \mathbf{M}^{i+1}\mathbf{e}^{\mathbf{0}}.$$

Intuitively unbounded growth in $\mathbf{e}^{\mathbf{i+1}}$ will occur if $||\mathbf{M}|| > 1$, where the operator $||\cdot||$ denotes the matrix norm[1]. The Lax-Richmeyr definition for stability states that

$$||\mathbf{M}|| \leq 1$$

is the necessary and sufficient condition for stability. A similar argument follows to derive the necessary and sufficient condition for stability for difference equation (3.20) which is

$$||\mathbf{\Sigma}|| \leq 1.$$

The Brennan and Schwartz condition for stability

Brennan and Schwartz [1978] pose that given $A+(1+B)+C = 1$ then $A \geq 0$, $B \geq 0$ and $C \geq 0$ is the condition for stability in the formulation

$$w_j^{i+1} = Aw_{j+1}^i + (1+B)w_j^i + Cw_{j-1}^i,$$

with the values A, B and C given by equation (3.10) and

$$\begin{aligned} \alpha &= \frac{1}{2}\sigma^2, \\ \beta &= r - \frac{1}{2}\sigma^2, \text{ and} \\ \gamma &= 0. \end{aligned}$$

This formulation is related to equation (3.9) but make some transpositions in order to force constant coefficients. The Brennen and Schwartz partial differential equation is otherwise perfectly suited for the general difference equation (3.19) with $\theta = 1$. The stability condition follows from the norm of the matrix $\mathbf{M}$. It can be shown that $||M||_1 = |A| + |B| + |C|$ and $||M||_2 = |A| + |B| + |C|$ and since $A + B + C = 1$, it follows that $||M||_1, ||M||_2 \geq 1$ if any of A, B or C is negative for any i or j. The Brennen and Schwartz condition is thus in this case consistent with the more general Lax and Richtmyer condition.

[1]For all compatible matrix norms it can be shown that $\rho(\mathbf{M}) \leq ||\mathbf{M}||$. Matrix norm here means $\min(L_1, L_2, L_\infty)$ where L_1, L_2 and L_∞ are the 1–norm, 2–norm and infinity–norms respectively

4.5.2 The Fourier analysis or von Neumann method to determine stability

In this section we adopt the analysis of Smith [1984]. Following is an abridged version. The Fourier analysis method expresses the initial values in terms of a finite Fourier series. It then considers the growth of a function that reduces this finite Fourier series for the initial time by a variables separable method. We formulate the Fourier series by making use of the complex exponential form, i.e.

$$\hat{f}_j^0 = \sum_{m=0}^{M} A_m e^{\mathbf{i}\varrho_m jh}, \qquad j = 0, 1, \ldots, M, \tag{4.14}$$

where A_m are unknown constants determined by the function $\hat{f}(q, s)$, $\mathbf{i} = \sqrt{-1}$, and $\varrho_m = m\pi/\ell$. The variable ℓ is the spatial interval over which the function is defined i.e. $Mh = \ell$.

The $M+1$ unknowns, $A_0, A_1, \ldots, A_M$ of equation (4.14) are solved with $M+1$ equations. Since the initial value equations are additive (we only consider linear difference equations), we only consider the propagation of a single initial value such as $e^{\mathbf{i}\varrho jh}$. The coefficients A_m are constant and therefore omitted; we thus investigate the propagation of the term $e^{\mathbf{i}\varrho jh}$ as τ increases.

We put

$$\hat{f}_j^i = e^{\mathbf{i}\varrho s} e^{\vartheta\tau} = e^{\mathbf{i}\varrho jh} e^{\vartheta ik} = e^{\mathbf{i}\varrho jh}\Upsilon^i, \tag{4.15}$$

where $\Upsilon = e^{\vartheta k}$ and ϑ is a complex constant.

The necessary and sufficient condition for stability is that

$$|\Upsilon| \leq \begin{cases} 1 & \text{If the exact solution of the difference equations does not increase exponentially with time, or} \\ 1 + \mathcal{O}(k) & \text{If the exact solution of the difference equations increases exponentially with time.} \end{cases} \tag{4.16}$$

4.5.3 Stability of the explicit, implicit, Crank & Nicolson and Douglas schemes

Schemes based on the initial value Black and Scholes equations

The fully explicit scheme: We evaluate the difference equation

$$\hat{f}_j^{i+1} = A\hat{f}_{j+1}^i + (1 + B)\hat{f}_j^i + C\hat{f}_{j-1}^i$$

by making use of the von Neumann analysis. Substituting the function $\hat{f}_j^i$ for $e^{\mathbf{i}\varrho jh}\Upsilon^i$ (equation 4.15) we obtain

$$\Upsilon^{i+1}e^{\mathbf{i}\varrho jh} = Ae^{\mathbf{i}\varrho(j+1)h}\Upsilon^i + (1+B)e^{\mathbf{i}\varrho jh}\Upsilon^i Ce^{\mathbf{i}\varrho(j-1)h}\Upsilon^i$$

By rearranging terms we obtain

$$\begin{aligned}\Upsilon &= A(\cos\varrho h + \mathbf{i}\sin\varrho h) + 1 + B + C(\cos\varrho h - \mathbf{i}\sin\varrho h)\\ &= \left[1+\gamma k + 2\alpha\frac{k}{h^2}(\cos(\varrho h)-1)\right] + \mathbf{i}\beta\frac{k}{h}\sin\varrho h\\ &= Y + \mathbf{i}X.\end{aligned}$$

For stability it is required that $|\Upsilon| \leq 0 \Rightarrow \sqrt{Y^2+X^2} \leq 0$. We evaluate the stability for values of ϱh:

$\varrho h = 0:$ $\sqrt{(1+\gamma k)^2} \leq 0$
resulting in the stability condition $-\frac{2}{k} \leq \gamma \leq 0$.

$\varrho h = \frac{\pi}{2}:$ $\sqrt{1+2\gamma k - 4\alpha\frac{k}{h^2} + \gamma^2k^2 - 4\alpha\gamma k\frac{k}{h^2} + 4\alpha^2\frac{k^2}{h^4} + \beta^2\frac{k^2}{h^2}} \leq 0$
Set $k = zh$ then
$\sqrt{1+2\gamma zh - 4\alpha\frac{z}{h} + \gamma^2z^2h^2 - 4\alpha\gamma z^2 + 4\alpha^2\frac{z^2}{h^2} + \beta^2z^2} \leq 1$
If we consider $\frac{1}{h^2}$ to be of $\mathcal{O}(1)$ then by approximation
$k \leq \frac{h^2}{2\alpha}$.

$\varrho h = \pi:$ $\sqrt{(1+\gamma k - 4\alpha\frac{k}{h^2})^2} \leq 1$
$\gamma k - 4\alpha\frac{k}{h^2} \leq 0$
resulting in the condition $h^2 \leq 4\frac{\alpha}{\gamma}$.

The fully implicit scheme: We evaluate the difference equation

$$-A\hat{f}_{j+1}^{i+1} + (1-B)\hat{f}_j^{i+1} - C\hat{f}_{j-1}^{i+1} = \hat{f}_j^i.$$

Substituting the function $\hat{f}_j^i$ for $e^{\mathbf{i}\varrho jh}\Upsilon^i$ we obtain

$$\begin{aligned}-Ae^{\mathbf{i}\varrho(j+1)h}\Upsilon^{i+1} + (1-B)e^{\mathbf{i}\varrho jh}\Upsilon^{i+1} - Ce^{\mathbf{i}\varrho(j-1)h}\Upsilon^{i+1} &= e^{\mathbf{i}\varrho jh}\Upsilon^i\\ -Ae^{\mathbf{i}\varrho h}\Upsilon + (1-B)\Upsilon - Ce^{-\mathbf{i}\varrho h}\Upsilon &= 1\\ \Upsilon\left\{1-B+(-A-C)\cos\varrho h + \mathbf{i}(-A+C)\sin\varrho h\right\} &= 1\end{aligned}$$

Rearranging terms yield

$$\begin{aligned}\Upsilon &= \frac{1}{1-\gamma k+2\alpha\frac{k}{h^2}(1-\cos\varrho h)+\mathbf{i}\beta\frac{k}{h}\sin\varrho h}\\ &= Y+\mathbf{i}X,\end{aligned}$$

where

$$\begin{aligned}Y &= \frac{1-\gamma k+2\alpha\frac{k}{h^2}(1-\cos\varrho h)}{\left(1-\gamma k+2\alpha\frac{k}{h^2}(1-\cos\varrho h)\right)^2+\beta^2\frac{k^2}{h^2}\sin^2\varrho h}, \text{ and}\\ X &= \frac{-\beta\frac{k}{h}\sin\varrho h}{\left(1-\gamma k+2\alpha\frac{k}{h^2}(1-\cos\varrho h)\right)^2+\beta^2\frac{k^2}{h^2}\sin^2\varrho h}.\end{aligned}$$

For stability it is required that $|\Upsilon| \leq 1 \Rightarrow \sqrt{Y^2+X^2} \leq 1$. We consider three cases, $\varrho h = 0$, $\varrho h = \frac{\pi}{2}$, and $\varrho h = \pi$, strictly for partial differential equations where $\gamma \leq 0$.

$\varrho h = 0$: $\frac{1}{1-\gamma k} \leq 1$
resulting in the condition $\gamma \leq 0$.

$\varrho h = \frac{\pi}{2}$: $\sqrt{\frac{\left(1-\gamma k+2\alpha\frac{k}{h^2}\right)^2+\beta^2\frac{k^2}{h^2}}{\left[\left(1-\gamma k+2\alpha\frac{k}{h^2}\right)^2+\beta^2\frac{k^2}{h^2}\right]^2}} \leq 1$
$-2\gamma k+4\alpha\frac{k}{h^2}+\gamma^2k^2-4\alpha\gamma\frac{k^2}{h^2}-4\alpha\frac{k^2}{h^4}+\beta^2\frac{k^2}{h^2} \geq 0$
If we take $\frac{1}{h^4}$ to be of $\mathcal{O}(1)$ and ignore higher orders, then by approximation $\frac{k}{h^2} \geq 0$.

$\varrho h = \pi$: $\sqrt{\left(\frac{1}{1-\gamma k+4\alpha\frac{k}{h^2}}\right)^2} \leq 1$
which leads to the condition $\gamma \leq 4\frac{\alpha}{h^2}$.

Since $\gamma \leq 0$ and $k > 0$ and $h > 0$ and generally $\alpha = \frac{1}{2}\sigma^2 s^2 > 0$, we conclude that the implicit scheme is always stable.

The Crank and Nicolson scheme: We evaluate the difference equation

$$A\hat{f}_{j+1}^{i+1}+B\hat{f}_j^{i+1}+C\hat{f}_{j-1}^{i+1}+A\hat{f}_{j+1}^{i}+B\hat{f}_j^{i}+C\hat{f}_{j-1}^{i}-2\hat{f}_j^{i+1}+2\hat{f}_j^{i}=0.$$

By substituting $\hat{f}_j^i$ for $e^{\mathbf{i}\varrho jh}\Upsilon^i$, we obtain after some algebra

$$\begin{aligned}\Upsilon &= \frac{-\gamma k-2\alpha\frac{k}{h^2}(\cos\varrho h-1)-2-\mathbf{i}\beta\frac{k}{h}\sin\varrho h}{\gamma k+2\alpha\frac{k}{h^2}(\cos\varrho h-1)-2+\mathbf{i}\beta\frac{k}{h}\sin\varrho h}\\ &= \frac{(-Y-2)-\mathbf{i}X}{(Y-2)+\mathbf{i}X}\end{aligned}$$

The absolute value of Υ is given by

$$|\Upsilon| = \frac{\sqrt{Y^2(Y^2-8)+X^2(X^2+8)+2(X^2Y^2+8)}}{4-4Y+Y^2+X^2},$$

where

$$\begin{aligned} Y &= \gamma k + 2\alpha\frac{k}{h^2}(\cos\varrho h - 1) \\ X &= \beta\frac{k}{h}\sin\varrho h. \end{aligned}$$

We evaluate three cases, namely

ϱ**h**	**Y**	**X**
0	γk	0
$\frac{\pi}{2}$	$\gamma k - 2\alpha\frac{k}{h^2}$	$\beta\frac{k}{h}$
π	$\gamma k - 4\alpha\frac{k}{h^2}$	0

Entering the inequality $\Upsilon \leq 1$ with additional constraints $\gamma \leq 0$, $\alpha \geq 0$, and $\beta \geq 0$ in Mathematica[2], the following results were obtained:

Reduce[
$\left(\sqrt{\left(\gamma k-2\alpha\left(\frac{k}{h^2}\right)\right)^2\left(\left(\gamma k-2\alpha\left(\frac{k}{h^2}\right)\right)^2-8\right)+\left(\beta\left(\frac{k}{h}\right)\right)^2\left(\left(\beta\left(\frac{k}{h}\right)\right)^2+8\right)+2\left(\left(\beta\left(\frac{k}{h}\right)\right)^2\left(\gamma k-2\alpha\left(\frac{k}{h^2}\right)\right)^2+8\right)}\right)\Big/$
$\left(4-4\left(\gamma k-2\alpha\left(\frac{k}{h^2}\right)\right)+\left(\gamma k-2\alpha\left(\frac{k}{h^2}\right)\right)^2+\left(\beta\left(\frac{k}{h}\right)\right)^2\right) \leq 1\&\&\gamma \leq 0\&\&$
$\alpha \geq 0\&\&\beta \geq 0\&\&k > 0\&\&h > 0, \{k, h\}]$

$\gamma \leq 0 \wedge \beta \geq 0 \wedge \alpha \geq 0 \wedge k > 0 \wedge h > 0$

Reduce $\left[\frac{\sqrt{\left(\gamma k-4\alpha\left(\frac{k}{h^2}\right)\right)^2\left(\left(\gamma k-4\alpha\left(\frac{k}{h^2}\right)\right)^2-8\right)+16}}{4-4\left(\gamma k-4\alpha\left(\frac{k}{h^2}\right)\right)+\left(\gamma k-4\alpha\left(\frac{k}{h^2}\right)\right)^2} \leq 1\&\&\gamma \leq 0\&\&\right.$
$\left.\alpha \geq 0\&\&\beta \geq 0\&\&k > 0\&\&h > 0, \{k, h\}\right]$

$\gamma \leq 0 \wedge \beta \geq 0 \wedge \alpha \geq 0 \wedge k > 0 \wedge h > 0$

Reduce $\left[\left\{\frac{\sqrt{(\gamma k)^2((\gamma k)^2-8)+16}}{4-4\gamma k+(\gamma k)^2} \leq 1\&\&\gamma \leq 0\&\&\alpha \geq 0\&\&\beta \geq 0\&\&k > 0\&\&h > 0\right\}, \gamma\right]$

$\beta \geq 0 \wedge \alpha \geq 0 \wedge k > 0 \wedge h > 0 \wedge \gamma \leq 0$

From these results it is evident that the Crank and Nicolson scheme is unconditionally stable.

[2] Wolfram Mathematica version 6.

Schemes based on the heat equation

The stability conditions for schemes based on the heat equation are well known (see for instance Smith [1984]). Both the implicit scheme and Crank–Nicolson schemes are unconditionally stable. The explicit scheme has a stability condition [Seydel, 2004] of

$$0 < \lambda \leq \frac{1}{2}.$$

4.6 Conclusion

We conclude this chapter with a summary of the characteristics of the different schemes:

Property	**Black–Scholes equation**	**Heat equation**
Truncation error		
Fully explicit	$\mathcal{O}(k) + \mathcal{O}(h^2)$	$\mathcal{O}(\kappa) + \mathcal{O}(\eta^2)$
Fully implicit	$\mathcal{O}(k) + \mathcal{O}(h^2)$	$\mathcal{O}(\kappa) + \mathcal{O}(\eta^2)$
Crank and Nicolson	$\mathcal{O}(k^2) + \mathcal{O}(h^2)$	$\mathcal{O}(\kappa^2) + \mathcal{O}(\eta^2)$
Douglas	NA	$\mathcal{O}(\kappa^2) + \mathcal{O}(\eta^4)$
Consistency		
Fully explicit	Consistent	Consistent
Fully implicit	Consistent	Consistent
Crank and Nicolson	Consistent	Consistent
Douglas	NA	Consitent
Stability		
Fully explicit	$-\frac{2}{k} \leq \gamma \leq 0, \quad k \leq \frac{h^2}{2\alpha}, \quad h^2 \leq 4\frac{\alpha}{\gamma}$	$0 < \lambda \leq \frac{1}{2}$
Fully implicit	$k > 0, \quad h > 0$	$\lambda > 0$
Crank and Nicolson	$k > 0, \quad h > 0$	$\lambda > 0$
Douglas	NA	$\lambda > 0$

Table 4.1: Summary of the properties of the most common finite difference schemes.

Chapter 5

Themes of the Du Fort and Frankel finite difference scheme

5.1 Introduction

Two important themes sprouted from the analysis of the early explicit finite difference scheme. The first was to improve the accuracy of the scheme while the second dealt with the conditional stability property of the fully explicit scheme. While implicit schemes effectively eliminate stability issues of the fully explicit scheme, explicit schemes still have desirable properties applicable to a number of problems that often occur in financial engineering problems. We investigate three such schemes. The first is a scheme recently suggested by Duffy [2006a] referred to as the MADE (modified alternating directional explicit) scheme. The MADE scheme sacrifices accuracy in order to obtain stability. The second scheme is known as the Richardson scheme. The Richardson scheme (see [Smith, 1984]) obtains its temporal derivative by central differences, thereby achieving a truncation error of

$$T_{\text{Richardson}} = \mathcal{O}(h^2) + \mathcal{O}(k^2).$$

This scheme proves to be unconditionally unstable, but a number of important observations lead to the third scheme and main subject of this document.
The Du Fort and Frankel scheme [Du Fort and Frankel, 1953], first published in 1953 improves on the Richardson scheme [Smith, 1984]. Du Fort and Frankel makes adjust-

ments to the diffusion term which results in a scheme that is both stable and explicit. Although stability is easily obtained by making a scheme implicit, the explicit property of the Du Fort and Frankel scheme is useful for many problems that occur in the field of financial engineering. This chapter briefly explores the Du Fort and Frankel scheme and derive a number of variations of occurring themes of this scheme that will be applied to financial engineering problems.

5.2 The MADE scheme

The MADE scheme was recently suggested by Duffy [2006a]. The scheme approximates the diffusion term of equation (2.5) by

$$\frac{\partial^2 f_j^i}{\partial s^2} \approx \frac{1}{h^2}(f_{j+1}^i - 2f_j^{i+1} + f_{j-1}^i).$$

The resulting difference equation is given by

$$0 = -\frac{1}{k}(f_j^{i+1} - f_j^i) + \frac{\alpha}{h^2}(f_{j+1}^i - 2f_j^{i+1} + f_{j-1}^i) + \frac{\beta}{2h}(f_{j+1}^i - f_{j-1}^i) + \gamma f_f^i$$

Rearrangement of terms give

$$f_j^{i+1} = \bar{A}f_{j+1}^i + \bar{B}f_j^i + \bar{C}f_{j-1}^i, \tag{5.1}$$

where

$$\begin{aligned} \bar{A} &= \frac{2\alpha k + \beta k h}{2h^2 + 4\alpha k}, \\ \bar{B} &= \frac{h^2 + \gamma k h^2}{h^2 + 2\alpha k}, \text{ and} \\ \bar{C} &= \frac{2\alpha k - \beta k h}{2h^2 + 4\alpha k}. \end{aligned}$$

5.2.1 Truncation error of the MADE scheme

The local truncation error is given by

$$\begin{aligned} T_j^i = G_j^i(F) &= -\frac{1}{k}(F_j^{i+1} - F_j^i) + \frac{\alpha}{h^2}(F_{j+1}^i - 2F_j^{i+1} + F_{j-1}^i) + \frac{\beta}{2h}(F_{j+1}^i - F_{j-1}^i) + \gamma F_j^i \\ &= -\frac{\partial F_j^i}{\partial q} + \alpha\frac{\partial^2 F_j^i}{\partial s^2} + \beta\frac{\partial F_j^i}{\partial s} + \gamma F_j^i + \dots \\ &\quad + k\left(-\frac{1}{2}\frac{\partial^2 F_j^i}{\partial q^2}\right) + \frac{k}{h^2}\left(-2\alpha\frac{\partial F_j^i}{\partial q}\right) + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F_j^i}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3 F_j^i}{\partial s^3}\right) + \dots \end{aligned}$$

Since F is the solution of the partial differential equation, the local truncation error is

$$\begin{aligned} T_j^i &\approx k\left(-\frac{1}{2}\frac{\partial^2 F_j^i}{\partial q^2}\right) + \frac{k}{h^2}\left(-2\alpha\frac{\partial F_j^i}{\partial q}\right) + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F_j^i}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3 F_j^i}{\partial s^3}\right) \\ &= \mathcal{O}(k) + \mathcal{O}(\frac{k}{h^2}) + \mathcal{O}(h^2). \end{aligned}$$

5.2.2 Consistency of the MADE scheme

The MADE scheme is only conditionally consistent with the initial value Black and Scholes partial differential equation. Assume there exists a relation between k and h such that

$$k = h^x.$$

The local truncation error may then be written as

$$T_j^i = h^x\left(-\frac{1}{2}\frac{\partial^2 F_j^i}{\partial q^2}\right) + h^{x-2}\left(-2\alpha\frac{\partial F_j^i}{\partial q}\right) + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F_j^i}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3 F_j^i}{\partial s^3}\right) + \ldots$$

By letting $h \to 0$ we observe that the second term

$$\lim_{h\to 0} h^{x-2}\left(-2\alpha\frac{\partial F_j^i}{\partial q}\right) \to 0, \quad x > 2, \tag{5.2}$$

doesn't unconditionally tend to zero. The MADE scheme is only consistent with the initial value Black and Scholes partial differential equation when $x > 2$.

Clearly the stated relationship between k and h doesn't guarantee $x > 2$ unconditionally. From the algebraic manipulation

$$x = \frac{\log k}{\log h},$$

it is clear that $k = 1$, for instance never yields a value for $x > 2$. If $k > 1$ then one would be tempted to choose values of $h \to 1^+$, while values of $k < 1$, as is mostly the case, values of $h \to 1^-$ appears tempting. This is depicted in figure (5.1). Before choosing a value of $h \approx 1$, it should be noted that the restriction of $x > 2$ was derived in the limit where $h \to 0$. Choosing $h \approx 1$ might thus not be the appropriate choice. The exact relationship between k and h for any value of h is not trivially derivable as the two quantities have different units. Perhaps a better way to determine this relationship is to formulate the problem in terms of number of grid points, which is unit-less for both the temporal and spatial axes. Let

$$M = N^x.$$

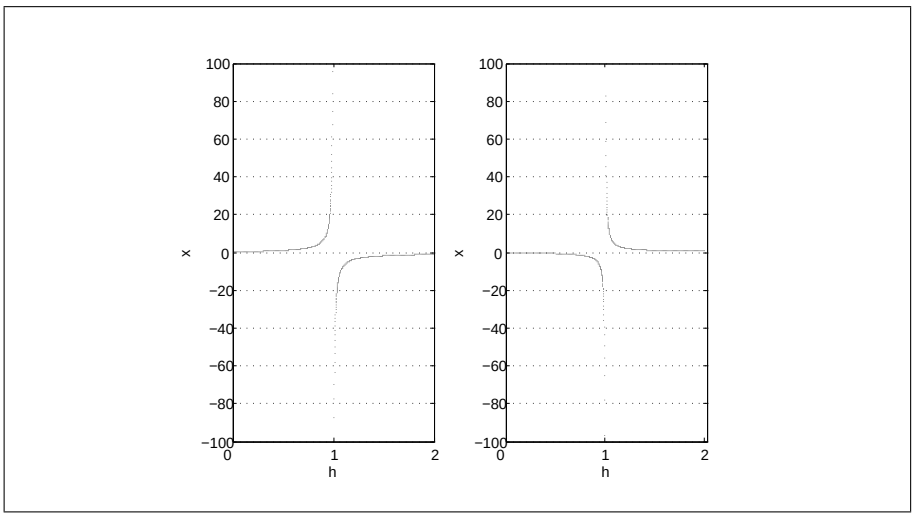

Figure 5.1: Values of x for an arbitrarily chosen $k < 1$ (left) and $k > 1$ (right). It is apparent that values of $h \to 1^-$ in the case where $k < 1$ or $h \to 1^+$ when $k > 1$ result in large positive values for x.

Taking $s_\chi = 0$, the truncation error is given by

$$T_j^i = TN^{-x}\left(-\frac{1}{2}\frac{\partial^2 F_j^i}{\partial q^2}\right) - \frac{2\alpha T}{s_\psi^2}N^{2-x}\frac{\partial F_j^i}{\partial q} + \frac{s_\psi}{N^2}\left(\frac{1}{12}\alpha\frac{\partial^4 F_j^i}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3 F_j^i}{\partial s^3}\right) + \dots$$

By letting $N \to \infty$ we find $T_j^i \to 0$ when $x > 2$. Since M and N are both without units and $M, N > 1$, we generalize that the scheme is consistent for $M > N^2$, which is severely restrictive.

5.2.3 Stability of the MADE scheme

We test stability of the MADE scheme by making use of the von Neumann technique. Substituting f_j^i from equation (5.1) for $e^{\mathbf{i}\varrho jh}\Upsilon^i$ (see equation 4.15) we obtain

$$e^{\mathbf{i}\varrho jh}\Upsilon^{i+1} = \bar{A}e^{\mathbf{i}\varrho j+1h}\Upsilon^i + \bar{B}e^{\mathbf{i}\varrho jh}\Upsilon^i + \bar{C}e^{\mathbf{i}\varrho j-1h}\Upsilon^i$$

Which, after rearranging terms gives

$$\begin{aligned}\Upsilon &= \bar{A}e^{\mathbf{i}\varrho h} + \bar{B} + \bar{C}e^{-\mathbf{i}\varrho h}\\ &= (\bar{A} + \bar{C})\cos\varrho h + B + \mathbf{i}(\bar{A} - \bar{C})\sin\varrho h,\\ \text{simplified as } \Upsilon &= X + \mathbf{i}Y,\end{aligned}$$

where

$$X = \frac{2\alpha k\cos\varrho h + h^2 + \gamma k h^2}{h^2 + 2\alpha k}, \text{ and} \qquad Y = \frac{\beta k h\sin\varrho h}{h^2 + 2\alpha k}.$$

In order to simplify our analysis we make the following assumptions:

- $s \in [s_\chi, s_\psi]$, where $s_\chi = 0$ and $s_\psi = Nh$.
- We define terms "small s" and "large s" as h and Nh respectively.
- We restrict our analysis to the classical Black and Scholes partial differential equation, namely $\alpha = \frac{1}{2}\sigma^2 s^2$, $\beta = rs$, and $\gamma = -r$.
- We assume $0 < rk < 1$, and
- We assume that N is sufficiently large so that $N - 1 \approx N \approx N + 1$.

Employing these assumptions, we obtain after algebraic manipulation the following simplifications for X and Y

$$X = \begin{cases} \frac{\sigma^2 k \cos \varrho h + (1 - rk)}{\sigma^2 k + 1} & \text{if } s \text{ is small, and} \\ \frac{\sigma^2 N^2 k \cos \varrho h + (1 - rk)}{\sigma^2 N^2 k + 1} & \text{if } s \text{ is large.} \end{cases}$$

and

$$Y = \begin{cases} \frac{rk \sin \varrho h}{\sigma^2 k + 1} & \text{if } s \text{ is small, and} \\ \frac{rNk \sin \varrho h}{\sigma^2 N^2 k + 1} & \text{if } s \text{ is large.} \end{cases}. \tag{5.3}$$

If s is small then both $X < 1$ and $Y < 1$. The most likely range for ϱh where $|\Upsilon| > 1$ is where $\sin \varrho h = \cos \varrho h = \sqrt{2}/2$. Under these conditions, values for $|\Upsilon|$ are given as

$$|\Upsilon| \approx \begin{cases} \frac{\frac{\sqrt{2}}{2}\sqrt{\sigma^4 k^2 + 1}}{\sigma^2 k + 1} < 1 & \text{if } rk \to 1, \\ \frac{\frac{\sqrt{2}}{2}\sigma^2 k + 1}{\sigma^2 k + 1} < 1 & \text{if } rk \to 0. \end{cases}.$$

The scheme seems stable when s is small

When s is large, i.e. $s = Nh$ coupled with $\sin \varrho h = \cos \varrho h = \frac{\sqrt{2}}{2}$ and $rk \to 1$, the approximate value for $|\Upsilon|$ that transpires is

$$|\Upsilon| \approx \frac{\frac{\sqrt{2}}{2}\sqrt{(\sigma^2 N^2 k)^2 + N^2}}{\sigma^2 N^2 k + 1}.$$

Under the assumed conditions, the stability of the scheme depends on the magnitude of $\sigma^2 k$ (for simplicity assumed to range between 0 and 1),

$$|\Upsilon| \approx \begin{cases} \frac{\sqrt{2}}{2N} < 1 & \text{if } \sigma^2 k \to 1, \\ \frac{\sqrt{2}N}{2} > 1 & \text{if } \sigma^2 k \to 0. \end{cases}.$$

The scheme may become unstable if both $\sigma^2 k \to 0$ and $rk \to 1$. For any given value for k, instability becomes likely if σ^2 becomes small in comparison to r. The likelihood

of an unstable scheme scales with the number of spatial steps. Since k is generally very small, interest rates need to be excessive coupled with very low volatility. Neither these normally occur and for virtually all practical applications, the MADE scheme may be regarded as stable.

An interesting observation is that stability for the MADE scheme is related to the relative magnitudes of r and σ^2. This theme often comes to the fore in discussions relating to spurious oscillations that occur in convection dominant partial differential equations [see for instance Duffy, 2004a; Seydel, 2004]. If convection is absent, then the MADE scheme becomes unconditionally stable.

Stability of the MADE scheme in the absence of convection

The MADE scheme for a heat equation is given by

$$\frac{v_j^{i+1} - v_j^i}{\kappa} = \frac{v_{j+1}^i - 2v_j^{i+1} + v_{j-1}^i}{\eta^2}$$

Rearranging terms give

$$v_j^{i+1} = \frac{\kappa}{\eta^2 + 2\kappa} v_{j+1}^i + \frac{\eta^2}{\eta^2 + 2\kappa} v_j^i + \frac{\kappa}{\eta^2 + 2\kappa} v_{j-1}^i.$$

By applying von Neumann analysis, we substitute v_j^i for $e^{\mathbf{i}\varrho j\eta}\Upsilon^i$ in order to obtain

$$\begin{aligned} e^{\mathbf{i}\varrho j\eta}\Upsilon^{i+1} &= \frac{\kappa}{\eta^2 + 2\kappa} e^{\mathbf{i}\varrho(j+1)\eta}\Upsilon^i + \frac{\eta^2}{\eta^2 + 2\kappa} e^{\mathbf{i}\varrho j\eta}\Upsilon^i + \frac{\kappa}{\eta^2 + 2\kappa} e^{\mathbf{i}\varrho(j-1)\eta}\Upsilon^i \\ \text{with } \Upsilon &= \frac{\kappa}{\eta^2 + 2\kappa} e^{\mathbf{i}\varrho\eta} + \frac{\eta^2}{\eta^2 + 2\kappa} + \frac{\kappa}{\eta^2 + 2\kappa} e^{-\mathbf{i}\varrho\eta} \\ &= 2\frac{\kappa}{\eta^2 + 2\kappa}\cos\varrho\eta + \frac{\eta^2}{\eta^2 + 2\kappa} \\ &= \frac{\eta^2 + 2\kappa\cos\varrho\eta}{\eta^2 + 2\kappa} \end{aligned}$$

which leads to $|\Upsilon| \leq 1$.

The possibility of instability in the MADE scheme is therefore only present for partial differential equations with convection terms.

5.2.4 An effective range for the MADE scheme

Figure (5.2) depicts the impact of inconsistency and instability on option prices. A European vanilla option was priced with $s_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.25$. The analytical value, using the generalized Black and Scholes closed form

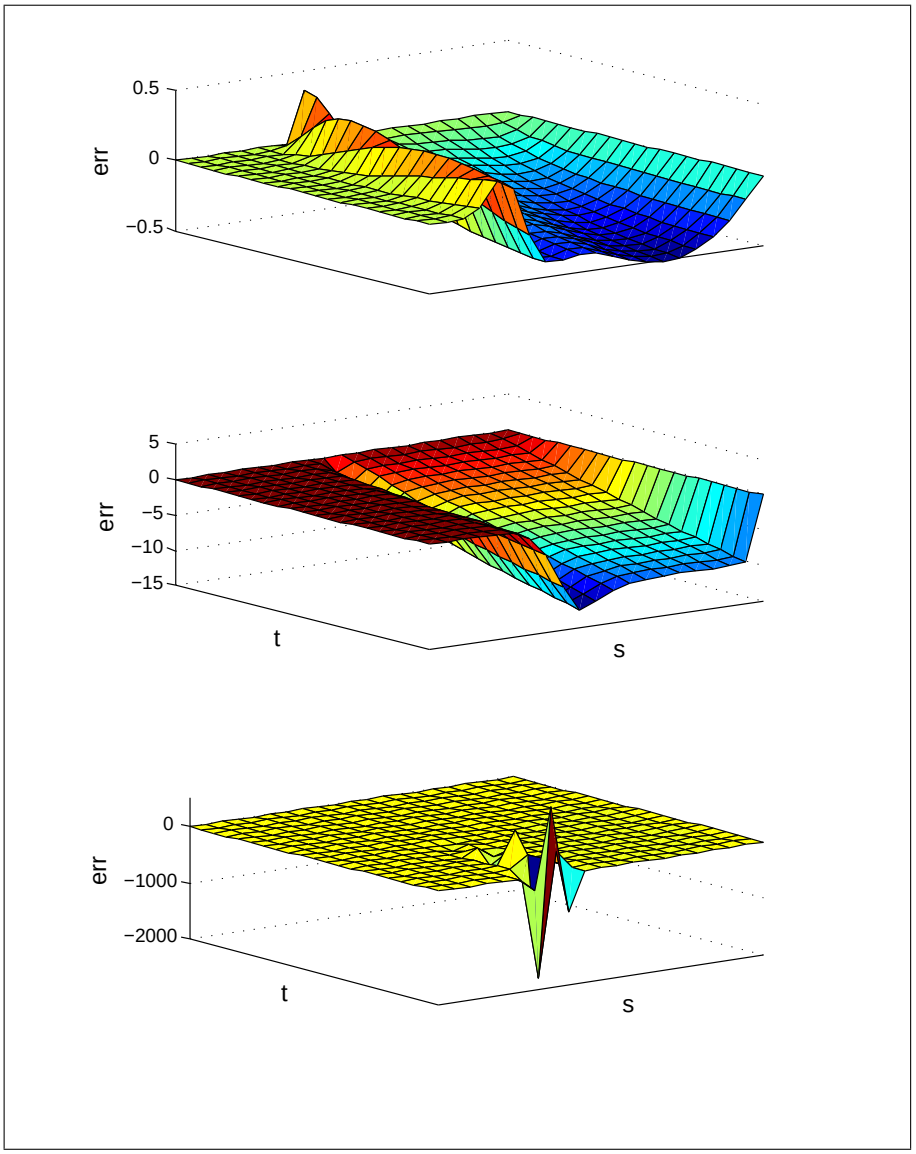

Figure 5.2: MADE: European option prices compared to analytical solution. Inputs were $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.25$. Stability and consistency (top) is enforced with $N = 20$ and $M = 440$. Inconsistency (middle) is apparent with $N = 220$ and $M = 40$, while the scheme also becomes unstable with $\sigma = 0.05$ and $r = 0.75$ (bottom).

formula [Haug, 1998], is 14.98.

In each case a similar total number of grid points were used (8800 grid points), and the analytical solution was subtracted from the finite difference solution. The top graphic shows a stable and consistent scheme achieved by putting $M > N^2$, namely $N = 20$ and $M = 440$. The error ranged between $-0.4 <$ err < 0.2. The middle graphic shows the impact of inconsistency with $N = 220$ and $M = 40$. The error ranged between $-12.5 <$ err < 0. The bottom graphic shows the impact of instability. This was achieved by using a small σ^2 compared to r, in this case $\sigma = 0.05$ and $r = 0.75$. The spatial and temporal steps were kept at $N = 220$ and $M = 40$. It must be noted that without the grid deliberately set to be inconsistent, the scheme appears to be remarkably stable in the sense that errors are bounded (results may by unusable though). With the "consistent grid" the scheme was still stable with $r = 10$ and $\sigma = 0.01$!

This fact is more a concern than an relief. In the case of the conventional schemes, especially the explicit scheme, there almost is no middle ground. Either the scheme is usable or it is not. The MADE scheme on the other hand may appear valid, but in reality it returns very poor estimates unless extreme care is taken.

5.2.5 Concluding remarks for the MADE scheme

Even though the MADE scheme may be the most unpractical scheme discussed to far, analysis of the scheme reveal some important themes that are also used by the Du Fort and Frankel scheme, the main object of our attention. The MADE scheme is remarkably stable, even though only one minute change was made to the explicit scheme. By changing the diffusion term one effectively reduce the values of the elements on the diagonal of the matrix $\mathbf{M_R}$ in the equation

$$\mathbf{M_L f^{i+1}} = \mathbf{M_R f^i} + \tilde{\mathbf{b}}.$$

By loosely referring to the absolute values of the elements on the diagonal as its *mass*, we observe the the theme of shifting mass to the left hand side also occurs in implicit schemes. The mass is either reduced to 0 in the case of the fully implicit scheme, or roughly half its weight shifts to the left hand side of the equation in the case of the Crank and Nicolson scheme. Whether this is a general rule of thumb remains a topic for further research, but the notion certainly have merit and empirically evidence

seems to support it. If this is the case then various techniques to force explicit schemes to become stable may present themselves.

The drawback of manipulating the diffusion term is that the central point has to be estimated over the temporal axis causing error terms of the form k^x/h^y. Such error terms will necessarily lead to inconsistencies, which will also become apparent with the Du Fort and Frankel scheme.

5.3 The Richardson scheme

We briefly provide background to the Richardson scheme. We do this for the initial value Black and Scholes partial differential equation, relying on sources such as Smith [1984] for findings relating to the heat equation. For our analysis we assume constant parameters.

5.3.1 Difference equation for the Richardson scheme

The Richardson scheme is similar to the fully explicit scheme, but instead of using the one–sided forward difference estimate of this scheme, the Richardson scheme makes use of central differences in order to estimate the temporal derivative, i.e. in the Black–Scholes framework

$$\frac{\partial f_j^i}{\partial q} \approx \frac{f_j^{i+k} - f_j^{i-k}}{2k}.$$

This adjustment leads to a difference equation

$$\begin{aligned} \hat{f}_j^{i+1} &= \hat{f}_{j+1}^i \left(\frac{2\alpha k}{h^2} + \frac{\beta k}{h} \right) + \hat{f}_j^i \left(\frac{-4\alpha k}{h^2} + 2\gamma k \right) + \hat{f}_{j-1}^i \left(\frac{2\alpha k}{h^2} - \frac{\beta k}{h} \right) + \hat{f}_j^{i-1} \\ &= A^* \hat{f}_{j+1}^i + B^* \hat{f}_j^i + C^* \hat{f}_{j-1}^i + \hat{f}_j^{i-1}. \end{aligned} \tag{5.4}$$

5.3.2 Local truncation error of the Richardson scheme

The local truncation error is given by

$$\begin{aligned} T_j^i = G_j^i(F) &= -\frac{1}{2k}(F_j^{i+1} - F_j^{i-1}) + \frac{\alpha}{h^2}(F_{j+1}^i - 2F_j^i + F_{j-1}^i) + \frac{\beta}{2h}(F_{j+1}^i - F_{j-1}^i) + \gamma F_j^i \\ &= -\frac{\partial F_j^i}{\partial q} + \alpha \frac{\partial^2 F_j^i}{\partial s^2} + \beta) \frac{\partial F_j^i}{\partial s} + \gamma F_j^i \ldots \\ &\quad -\frac{1}{3}k^2 \frac{\partial^3 F_j^i}{\partial q^3} + \ldots + \frac{\alpha}{12} h^2 \frac{\partial^4 F_j^i}{\partial s^4} + \ldots + \frac{\beta}{6} h^2 \frac{\partial^3 F_j^i}{\partial s^3} + \ldots \end{aligned}$$

Since F is the solution to the partial differential equation we have the principal part of the truncation error is thus

$$T_j^i = -\frac{1}{3}k^2\frac{\partial^3 F_j^i}{\partial q^3} + \frac{1}{12}h^2\left(2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\partial^4 F_j^i}{\partial s^4}\right).$$

Hence

$$T_j^i = \mathcal{O}(k^2) + \mathcal{O}(h^2).$$

5.3.3 Consistency of the Richardson scheme

If we set $k = zh$ then it can be shown that

$$\begin{aligned}\lim_{h\to 0} T_j^i &= \lim_{h\to 0} -\frac{1}{3}k^2\frac{\partial^3 F_j^i}{\partial q^3} + \frac{1}{12}h^2\left(2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\partial^4 F_j^i}{\partial s^4}\right)\\ &= \lim_{h\to 0} -\frac{1}{3}z^2h^2\frac{\partial^3 F_j^i}{\partial q^3} + \frac{1}{12}h^2\left(2\frac{\partial^3 F_j^i}{\partial s^3} + \frac{\partial^4 F_j^i}{\partial s^4}\right)\\ &= 0.\end{aligned}$$

The Richardson scheme is therefore consistent with the initial value Black–Scholes partial differential equation.

5.3.4 Stability of the Richardson scheme

In order to derive the stability conditions of this three time level scheme, we make use of a technique described by Smith [1984]. The difference equation (5.4) may be written in matrix form as

$$\begin{pmatrix}\hat{f}_1^{i+1}\\ \hat{f}_2^{i+1}\\ \vdots\\ \hat{f}_{N-2}^{i+1}\\ \hat{f}_{N-1}^{i+1}\end{pmatrix} = \begin{pmatrix}B^* & A^* & & & \\ C^* & & 0 & & \\ & \ddots & & & \\ & 0 & & & A^*\\ & & & C^* & B^*\end{pmatrix}\begin{pmatrix}\hat{f}_1^{i}\\ \hat{f}_2^{i}\\ \vdots\\ \hat{f}_{N-2}^{i}\\ \hat{f}_{N-1}^{i}\end{pmatrix} + \begin{pmatrix}\hat{f}_1^{i-1}\\ \hat{f}_2^{i-1}\\ \vdots\\ \hat{f}_{N-2}^{i-1}\\ \hat{f}_{N-1}^{i-1}\end{pmatrix} + \begin{pmatrix}C^*\hat{f}_0^i\\ 0\\ \vdots\\ 0\\ A^*\hat{f}_N^i\end{pmatrix},$$

or alternatively

$$\mathbf{f^{i+1}} = \mathbf{M}\mathbf{f^i} + \mathbf{f^{i-1}} + \tilde{\mathbf{b}}^{\mathbf{i}}, \tag{5.5}$$

where the symbols have a similar meaning than in equation (4.10). If we put

$$\mathbf{v^i} = \begin{pmatrix}\mathbf{f^i}\\ \mathbf{f^{i-1}}\end{pmatrix},$$

then equation (5.5) may be written as a two time level equation

$$\mathbf{v^{i+1}} = \mathbf{P}\mathbf{v^i} + \mathbf{c^i},$$

where

$$\mathbf{P} = \begin{pmatrix} \mathbf{M} & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix},$$

and

$$\mathbf{c^i} = \begin{pmatrix} \tilde{\mathbf{b}}^{\mathbf{i}} \\ 0 \end{pmatrix},$$

and $\mathbf{I}$ is the identity matrix of order $N-1$.

The difference scheme is stable if each of the the eigenvalues of the matrix $\mathbf{P}$ has an absolute value ≤ 1 [Smith, 1984]. By following a similar argument as described by Smith [1984, example 3.2], it can be shown that the eigenvalues ω of $\mathbf{P}$ are the eigenvalues of the matrix

$$\begin{pmatrix} \omega_g & 1 \\ 1 & 0 \end{pmatrix},$$

where ω_g is the g^{th} eigenvalue of matrix $\mathbf{M}$. In order to find the eigenvalues we evaluate

$$\det \begin{pmatrix} \omega_g - \omega & 1 \\ 1 & -\omega \end{pmatrix} = 0,$$

giving

$$\omega^2 - \omega_g \omega - 1 = 0.$$

The eigenvalues ω_g are given by (see Smith [1984])

$$\omega_g = A^* + 2\left(\sqrt{B^* C^*}\right)\cos\frac{g\pi}{N}, \qquad g = 1, 2, \ldots, N-1.$$

After substitution of A^*, B^* and C^* and some algebraic manipulation this becomes

$$\omega_g = \frac{k}{h^2}\left(2\alpha + \beta h + 2\cos\frac{g\pi}{N}(-8\alpha^2 + 4\alpha\beta h + 4\alpha\beta h^2 + 2\beta\gamma h^3)^{\frac{1}{2}}\right). \tag{5.6}$$

The eigenvalues of matrix $\mathbf{P}$ are

$$\omega = \frac{\omega_g \pm \sqrt{\omega_g^2 + 4}}{2}, \tag{5.7}$$

and the stability condition is

$$|\omega_g \pm \sqrt{\omega_g^2 + 4}| \leq 2.$$

In order to determine whether the Richardson scheme is stable, we assume a grid that increasingly becomes finer, i.e. $M, N \to \infty$ and $k, h \to 0$. We also evaluate the angle $\frac{g\pi}{N}$ where g is small implying that $\cos\frac{g\pi}{N} \to 1$.
We consider 3 relations between k and h:

Case h $> \sqrt{\mathbf{k}}$: $\omega_g \to \infty$ clearly results in $|\omega| > 1$ implying an unstable scheme.

Case h $= \sqrt{\mathbf{k}}$: $\omega_g \to 2\alpha + \mathbf{i}\sqrt{32}\alpha$. We write this as $X + \mathbf{i}Y$ with $X = 2\alpha$ and $Y = \sqrt{32}\alpha$. The stability condition is

$$|X + \mathbf{i}Y \pm \sqrt{(X + \mathbf{i}Y)^2 + 4}| \leq 2.$$

This may in turn be written as

$$\begin{aligned} |X + \mathbf{i}Y \pm V + \mathbf{i}W| &\leq 2 \\ \text{leading to } (X+V)^2 + (Y+W)^2 &\leq 4, \end{aligned}$$

where [Rabinowitz]

$$\begin{aligned} V &= \frac{1}{\sqrt{2}}\left(\sqrt{(X^2+4-Y^2)^2 + 4X^2Y^2} + X^2 + 4 - Y^2\right)^{\frac{1}{2}}, \text{ and} \\ W &= \frac{\text{sgn}(X^2+4-Y^2)}{\sqrt{2}}\left(\sqrt{(X^2+4-Y^2)^2 + 4X^2Y^2} - X^2 - 4 + Y^2\right)^{\frac{1}{2}}. \end{aligned}$$

The function sgn$(\cdot)$ is defined as

$$\text{sgn}(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x = 0, \\ -1 & \text{if} \quad x < 0. \end{cases}.$$

After some algebraic manipulation we obtain

$$V = \frac{1}{\sqrt{2}}\left(\sqrt{(28\alpha^2)^2 + 277\alpha^2 + 16} - 28\alpha^2 + 4\right)^{\frac{1}{2}}.$$

Since $\alpha > 0$ we note that the lowest possible value for V is where $\alpha \to 0$. The lower bound for V is therefore

$$V > 2.$$

Since $X > 0$ and $(Y + W)^2 > 0$ we conclude that

$$(X+V)^2 + (Y+W)^2 > 4,$$

hence the scheme is unstable for $h = \sqrt{k}$.

Case $\mathbf{h < \sqrt{k}}$: In this case $\omega_g \rightarrow 0$ which leads to $|\omega| \rightarrow 2$. Since $|\omega| = 2$ only occurs in the limit when $N, M \rightarrow \infty$ we conclude that for any finite grid $|\omega| > 2$ indicating an unstable scheme in the Lax Richtmyer sense.

5.3.5 Concluding remarks for the Richardson scheme

The Richardson scheme proposes a method whereby the local truncation error of the fully explicit scheme is improved so that the error of the approximating temporal derivative is of order $\mathcal{O}(k^2)$. Consistent with remarks by Smith [1984] (see 4.3) the additional accuracy is only obtained by adding an additional grid point in calculating the approximating difference equation. The Richardson scheme is unconditionally unstable. In line with the concluding remarks for the MADE scheme, certain observations relating to the Richardson scheme is relevant. Where the MADE scheme removes mass from the diagonal of the matrix $\mathbf{M_R}$ in the equation

$$\mathbf{M_L f^{i+1}} = \mathbf{M_R f^i} + \tilde{\mathbf{b}},$$

the Richardson scheme's main activity, namely its temporal estimation, occurs on the main diagonal, effectively adding weight the main diagonal.

5.4 The Du Fort and Frankel scheme

5.4.1 Introduction

The Du Fort and Frankel scheme, proposed in 1953 [Du Fort and Frankel, 1953], makes use of various themes discussed in the two above schemes. It came into existence in an effort to address the instability associated with the Richardson scheme [Smith, 1984]. The scheme is explicit, unconditionally stable and second order accurate in both the spatial and temporal dimensions. As is the case with the MADE scheme, the Du Fort and Frankel scheme is conditionally consistent with the partial differential equation it solves. Furthermore, since the Du Fort and Frankel scheme is a two step method, calculating the first temporal vector after the initial boundary requires some other method.

5.4.2 Difference equation for the Du Fort and Frankel scheme

The Du Fort and Frankel scheme makes use of a time derivative estimation similar to the Richardson scheme,

$$\frac{\partial f_j^i}{\partial q} \approx \frac{1}{2k}(f_j^{i+1} - f_j^{i-1}).$$

The diffusion term is estimated by

$$\frac{\partial^2 f_j^i}{\partial s^2} \approx \frac{1}{h^2}(f_{j+1}^i - (f_j^{i+1} + f_j^{i-1}) + f_{j-1}^i).$$

These estimates lead to a difference equation

$$0 = -\frac{1}{2k}(f_j^{i+1} - f_j^{i-1}) + \frac{\alpha}{h^2}(f_{j+1}^i - f_j^{i+1} - f_j^{i-1} + f_{j-1}^i) + \frac{\beta}{2h}(f_{j+1}^i - f_{j-1}^i) + \gamma f_j^i$$

After rearrangement of terms we obtain

$$f_j^{i+1} = \ddot{A} f_{j+1}^i + \ddot{B} f_j^i + \ddot{C} f_{j-1}^i + \ddot{D} f_j^{i-1},$$

where

$$\begin{aligned}
\ddot{A} &= \frac{2\alpha k + \beta k h}{h^2 + 2\alpha k}, \\
\ddot{B} &= \frac{2\gamma k h^2}{h^2 + 2\alpha k}, \\
\ddot{C} &= \frac{2\alpha k - \beta k h}{h^2 + 2\alpha k}, \text{ and} \\
\ddot{D} &= \frac{h^2 - 2\alpha k}{h^2 + 2\alpha k}.
\end{aligned}$$

5.4.3 Truncation error

The truncation error is given by

$$\begin{aligned}
T_j^i &= G_j^i(F) \\
&= -\frac{1}{2k}\left(F_j^i + k\frac{\partial F}{\partial q} + \frac{k^2}{2}\frac{\partial^2 F}{\partial q^2} + \frac{k^3}{3!}\frac{\partial^3 F}{\partial q^3} + \ldots - F_j^i + k\frac{\partial F}{\partial q} - \frac{k^2}{2}\frac{\partial^2 F}{\partial q^2} + \frac{k^3}{3!}\frac{\partial^3 F}{\partial q^3} + \ldots\right) \\
&\quad + \frac{\alpha}{h^2}\left(F_j^i + h\frac{\partial F}{\partial s} + \frac{h^2}{2}\frac{\partial^2 F}{\partial s^2} + \frac{h^3}{3!}\frac{\partial^3 F}{\partial s^3} + \ldots + F_j^i - h\frac{\partial F}{\partial s} + \frac{h^2}{2}\frac{\partial^2 F}{\partial s^2} - \frac{h^3}{3!}\frac{\partial^3 F}{\partial s^3} + \ldots\right) \\
&\quad + \frac{\alpha}{h^2}\left(-F_j^i - k\frac{\partial F}{\partial q} - \frac{k^2}{2}\frac{\partial^2 F}{\partial q^2} - \frac{k^3}{3!}\frac{\partial^3 F}{\partial q^3} + \ldots - F_j^i + k\frac{\partial F}{\partial q} - \frac{k^2}{2}\frac{\partial^2 F}{\partial q^2} + \frac{k^3}{3!}\frac{\partial^3 F}{\partial q^3} + \ldots\right) \\
&\quad + \frac{\beta}{2h}\left(F_j^i + h\frac{\partial F}{\partial s} + \frac{h^2}{2}\frac{\partial^2 F}{\partial s^2} + \frac{h^3}{3!}\frac{\partial^3 F}{\partial s^3} + \ldots - F_j^i - h\frac{\partial F}{\partial s} + \frac{h^2}{2}\frac{\partial^2 F}{\partial s^2} - \frac{h^3}{3!}\frac{\partial^3 F}{\partial s^3} + \ldots\right) \\
&\quad + \gamma F_j^i
\end{aligned}$$

After simplification this becomes

$$\begin{aligned} T_j^i &= -\frac{\partial F}{\partial q} + \alpha\frac{\partial^2 F}{\partial s^2} + \beta\frac{\partial F}{\partial s} + \gamma F \ldots \\ &\quad + k^2\left(-\frac{1}{3}\frac{\partial^3 F}{\partial q^3}\right) + h^2\left(\frac{1}{12}\alpha\frac{\partial^4 F}{\partial s^4} + \frac{1}{6}\beta\frac{\partial^3 F}{\partial s^3}\right) + \frac{k^2}{h^2}\left(-\alpha\frac{\partial^2 F}{\partial q^2}\right) + \ldots \end{aligned}$$

Since F is the solution to the partial differential equation, the principal error may be summarised as

$$T_j^i = \mathcal{O}(k^2) + \mathcal{O}(h^2) + \mathcal{O}\left(\frac{k^2}{h^2}\right).$$

Although the scheme is second order accurate in both the temporal and spatial dimension, comments by Lindsay [2005] suggests that increased accuracy can only be obtained by increasing the number of time and spatial steps in concert. This comment will be investigated in more detail later in this document.

5.4.4 Consistency

Since the principal truncation error has a term $\mathcal{O}(k^2/h^2)$, consistency will be conditional. Put

$$M = N^x.$$

Recalling that $k = T/M$ and $h = (s_\psi - s_\chi)/N$, the principal truncation error may then be written as

$$T_j^i \approx \frac{T^2}{N^{2x}}\left(-\frac{1}{3}\frac{\partial^3 F}{\partial q^3}\right) + \frac{(s_\psi - s_\chi)^2}{N^2}\left(\frac{1}{12}\alpha\frac{^4F}{s^4} + \frac{1}{6}\beta\frac{\partial^3 F}{\partial s^3}\right) + \frac{T^2 N^{2-2x}}{(s_\psi - s_\chi)^2}\left(-\alpha\frac{\partial^2 F}{\partial q^2}\right) ..$$

By making the grid increasingly finer we find that the error does only tend to zero if $x > 1$. Since $N, M > 1$, we generalize that the consistency condition holds when $M > N$.

$$\lim_{N\to\infty} T_j^i = \begin{cases} \infty & \text{if } x < 1, \\ \frac{T^2}{(s_\psi - s_\chi)^2}\left(-\alpha\frac{\partial^2 F}{\partial q^2}\right) & \text{if } x = 1, \text{ and} \\ 0 & \text{if } x > 1. \end{cases}$$

The scheme is consistent with the hyperbolic partial differential equation

$$\frac{\partial F}{\partial q} + \varpi\Omega = \alpha\frac{\partial^2 F}{\partial s^2} + \beta\frac{\partial F}{\partial s} + \gamma F,$$

where

$$\begin{aligned} \varpi &= \frac{\alpha T^2 N^{2-2x}}{(s_\psi - s_\chi)^2}, \text{ and} \\ \Omega &= \frac{\partial^2 F}{\partial q^2}. \end{aligned}$$

5.4.5 Impact of inconsistency on the accuracy of the Du Fort and Frankel scheme

The "additional" term, $\varpi\Omega$, in the partial differential equation impacts on the accuracy of the estimate. If we assume that $s_\chi \approx 0$ and noticing that ϖ reaches a maximum value when $\alpha = \frac{1}{2}\sigma^2 s_\psi^2$, the coefficient may be simplified to $\varpi = \frac{1}{2}\sigma^2 T^2 N^{2-2x}$. This quantity clearly scales inversely with x.

The second relevant quantity of $\varpi\Omega$ is the value of the partial derivative, which is difficult to assess for a general case. Instead, we approximate it on a case by case basis with the difference equation

$$\frac{\partial^2 F_j^i}{\partial q^2} \approx \frac{1}{k^2}\left(F_j^{i+1} - 2F_j^i + F_j^{i-1}\right),$$

where F_j^i is the true solution at grid points i and j. Graphic (5.3) shows the estimation of Ω for a European call option. Noticing the scale of the value axes, it is apparent that Ω becomes a factor, firstly close to maturity when time decay is rapid, and secondly along the present value of the strike of the option, where time decay is a maximum for any given time to maturity.

It is worthwhile to note that, unless the strike price is very high, ϖ is comparatively small when Ω is big, namely close to the strike. A simple analysis, assuming that the strike is $X = (s_\psi - s_\chi)/2$, and recalling that $N = (s_\psi - s_\chi)/h$, reveals a vastly reduced coefficient associated with high values for the second temporal derivative.

$$\varpi \approx \frac{\sigma^2 T^2 N^{1-2x}}{4h}.$$

An important factor to investigate is the impact of discontinuities on the value of Ω. Since F is the solution of the partial differential equation, any discontinuity may result in infinite partial derivatives. If such discontinuities occur near the maximum value of the spatial derivative, then possibly both ϖ and Ω may become very large resulting in significant inaccuracies. A good example of such behavior my occur with barrier options where the barrier level (H) is set high such that $H = s_\psi$. The value of Ω for such a barrier option is shown in figure (5.4). The value for Ω was numerically obtained from the closed form solution for barrier options by Merton and also Rubinstein [Haug, 1998]. The discontinuity at the barrier level results in high levels for Ω which in turn will be multiplied with a relative high value for ϖ. Unless M will be chosen to be high in comparison to N, the Du Fort and Frankel scheme may prove inefficient for

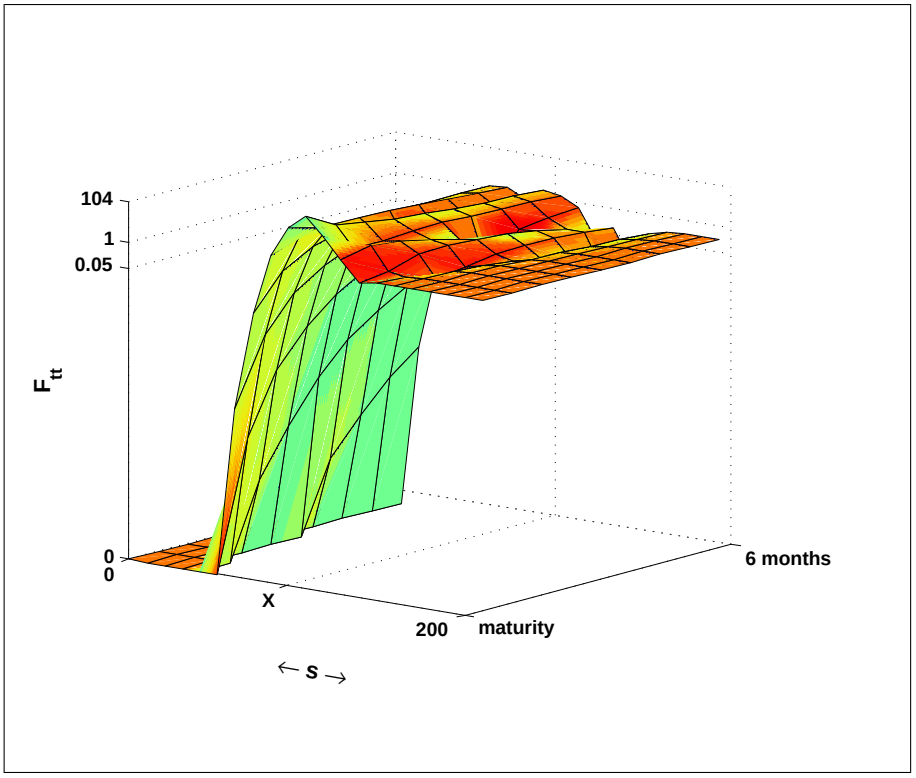

Figure 5.3: The numerical estimate for $\frac{\partial^2 F}{\partial q^2}$ for a European call option. Inputs were $s = 0 : 10 : 200$, $X = 100$, $T = 0 : 0.05 : 0.5$, $r = 0.1$ and $\sigma = 0.25$.

options where a high degree of discontinuity in the price occurs.

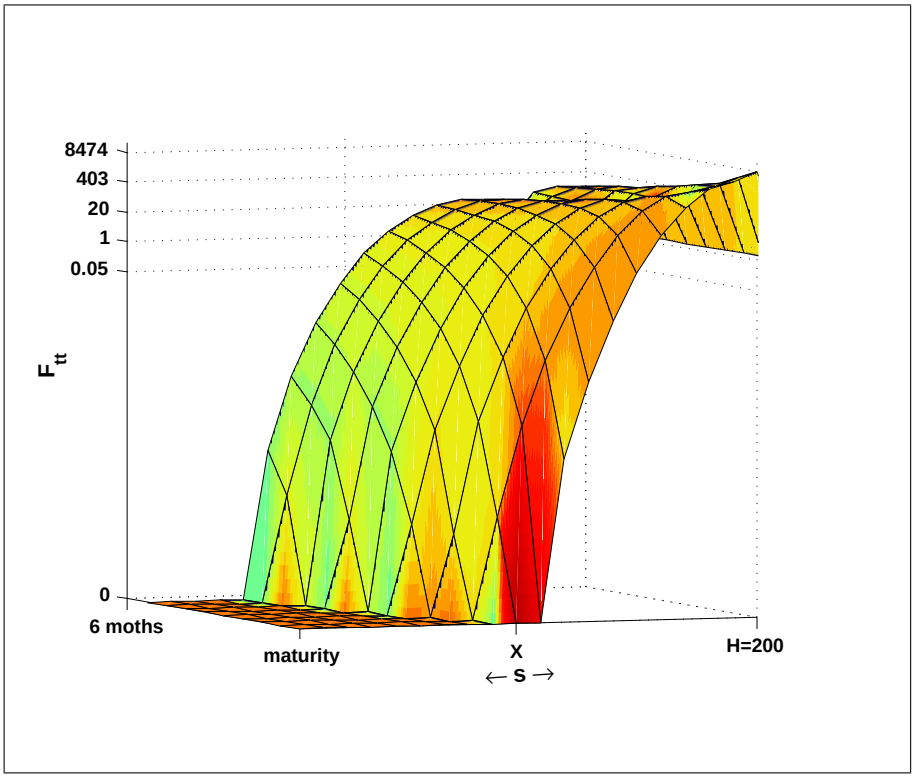

Figure 5.4: The numerical estimate for $\frac{\partial^2 F}{\partial q^2}$ for an up-and-out barrier option. Inputs were $s = 0 : 10 : 200$, $X = 100$, $T = 0 : 0.05 : 0.5$, $r = 0.1$ and $\sigma = 0.25$. The barrier is set to $H = s_\psi = 200$.

5.4.6 Stability of the Du Fort and Frankel scheme

The Du Fort and Frankel scheme is unconditionally stable. This result is well documented and often shown for pure diffusion parabolic partial differential equations [see for instance Smith, 1984]. However, the presence of convection in the Black Sholes equation challenges any attempt to proof stability. Following is an outline showing stability under extreme conditions.

The Du Fort and Frankel scheme may be presented in matrix form as

$$\mathbf{f^{i+1}} = \mathbf{M f^i} + \ddot{D}\mathbf{F^{i-1}} + \tilde{\mathbf{b}}^{\mathbf{i}},$$

where $\mathbf{f^x}$, $\mathbf{M}$ and $\tilde{\mathbf{b}}^{\mathbf{x}}$ have similar meanings to equation (5.5) above, and $\ddot{D}$ is scalar. Since the Du Fort and Frankel scheme involves more than one time step, our approach will be similar to that of the Richardson scheme above. Put

$$\mathbf{v^i} = \begin{pmatrix} \mathbf{f^i} \\ \mathbf{f^{i-1}} \end{pmatrix}, \qquad \mathbf{P} = \begin{pmatrix} \mathbf{M} & \ddot{D}\mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}, \text{ and} \qquad \mathbf{c^i} = \begin{pmatrix} \tilde{\mathbf{b}}^{\mathbf{i}} \\ 0 \end{pmatrix},$$

where $\mathbf{I}$ is the identity matrix. The difference scheme may then be rewritten as

$$\mathbf{v^{i+1}} = \mathbf{P}\mathbf{v^i} + \mathbf{c^i}.$$

We need to find the eigenvalues of the matrix

$$\begin{pmatrix} \omega_g & \ddot{D} \\ 1 & 0 \end{pmatrix},$$

where ω_g represents the $g^{\text{'th}}$ eigenvalue of the matrix $\mathbf{M}$. Similar to the argument in section (5.3.4) [see also Smith, 1984], the eigenvalues are given by

$$\omega = \frac{\omega_g \pm \sqrt{\omega_g^2 + 4D}}{2}. \tag{5.8}$$

The eigenvalues ω_g may be found in closed form

$$\begin{aligned} \omega_g &= \ddot{A} + 2\sqrt{\ddot{B}\ddot{C}}\cos\frac{g\pi}{N} \\ &= \frac{1}{h^2 + 2\alpha k}\left(2\alpha k + \beta k h + 2kh\sqrt{4\alpha\gamma - 2\beta\gamma h}\right)\cos\frac{g\pi}{N}. \end{aligned}$$

If we put $k = zh$, we may, after some basic algebraic manipulation, rewrite the above as

$$\omega_g = \frac{1}{h + 2\alpha z}\left(2\alpha z + \beta z h + 2zh\sqrt{4\alpha\gamma - 2\beta\gamma h}\right)\cos\frac{g\pi}{N}. \tag{5.9}$$

We investigate stability under two extreme cases, firstly for a grid that becomes increasingly fine, i.e. $h \to 0$, and secondly for a grid with only a few grid points, such that $h \to s_\psi$, $k \to T$ and $s \approx s_\psi$.

Stability with a fine grid

If $h \to 0$ then $\omega_g \to \cos\frac{g\pi}{N}$ and $\ddot{D} \to -1$. We consider 3 cases:

$\cos\frac{q\pi}{N} = 0$

$$\begin{aligned} \omega &= \mathbf{i} \\ \text{with absolute value } |\omega| &= 1 \end{aligned}$$

$\cos\frac{q\pi}{N} = 1$

$$\begin{aligned} \omega &= \frac{1 \pm \sqrt{-3}}{2} \\ &= \frac{1}{2} \pm \mathbf{i}\frac{\sqrt{3}}{2} \\ \text{with absolute value } |\omega| &= \sqrt{\frac{1}{4} + \frac{3}{4}} \\ &= 1 \end{aligned}$$

$\cos\frac{q\pi}{N} = -1$

$$\begin{aligned} \omega &= \frac{1 \pm \sqrt{-3}}{2} \\ &= \frac{1}{2} \pm \mathbf{i}\frac{\sqrt{3}}{2} \\ \text{with absolute value } |\omega| &= \sqrt{\frac{1}{4} + \frac{3}{4}} \\ &= 1 \end{aligned}$$

The Du Fort and Frankel scheme appears to be stable with an extremely fine grid.

Stability with a coarse grid

We investigate the impact on stability when the grid becomes coarser, i.e. $M, N \to 1$. This results in $k \to T$, $h \to s_\psi$ and $s = s\psi$. Substituting these into the coefficients $\ddot{A}$, $\ddot{B}$, $\ddot{C}$ and $\ddot{D}$ gives simplified values

$$\begin{aligned} \ddot{A} &= \frac{\sigma^2 T + rT}{\sigma^2 T + 1}, \\ \ddot{B} &= -\frac{2rT}{\sigma^2 T + 1}, \\ \ddot{C} &= \frac{\sigma^2 T - rT}{\sigma^2 T + 1}, \text{ and} \\ \ddot{D} &= \frac{1 - \sigma^2 T}{\sigma^2 T + 1}. \end{aligned}$$

Put

$$E = \ddot{A} + 2\sqrt{\ddot{B}\ddot{C}}.$$

We investigate cases where $r \geq \sigma^2$ and where $r < \sigma^2$.

1. $r \geq \sigma^2$

   We investigate two sub cases. The first is when $\sigma^2 \to 0$ and the second is when $\sigma^2 \to r$.

- $\sigma^2 \to 0$

$$E \to rT(1 + 2\sqrt{2}).$$

The majority of finance problems occur in the range $r < 1$. If $M > 1 + 2\sqrt{2}$, which is certainly the minimum number of temporal steps (given that at least 3 steps are required), then $E < 1$. If we assume an upper bound $E = 1$ and $|\cos g\pi| = 1$ then

$$\begin{aligned} \omega &= \frac{E \pm \sqrt{E^2 - 4D}}{2} \\ \omega &= \frac{1}{2} \pm \mathbf{i}\frac{\sqrt{3}}{2} \\ \text{with absolute value } |\omega| &= \sqrt{\frac{1}{4} + \frac{3}{4}} \\ &= 1. \end{aligned}$$

- $\sigma^2 \to r$

The coefficients will tend to

$$\begin{aligned} \ddot{A} &\to \frac{2rT}{1 + rT}, \\ \ddot{B} &\to -\frac{2rT}{1 + rT}, \\ \ddot{C} &\to 0, \\ \ddot{D} &\to \frac{1 - rT}{1 + rT}. \end{aligned}$$

Substituting these values into E gives $E = \ddot{A}$. If we assume that $rT < 1$, which certainly is the case for the vast majority of financial problems, then $E \leq 1$. Assume the upper bound $E = 1$ and with a similar argument than above it can be shown that

$$|\omega| \leq 1.$$

2. $r \leq \sigma^2$

Since $r \to \sigma^2 \Leftrightarrow \sigma^2 \to r$, we only investigate case where $r \to 0$. Adjusting the parameters for these assumptions leads to

$$\begin{aligned} \ddot{A} &\to \frac{\sigma^2 T}{1 + \sigma^2 T}, \\ \ddot{B} &\to 0, \\ \ddot{C} &\to \frac{\sigma^2 T}{1 + \sigma^2 T}, \text{ and} \\ \ddot{D} &\to \frac{1 - \sigma^2 T}{1 + \sigma^2 T}. \end{aligned}$$

By following a similar argument than above, we find

$$E = \ddot{A} < 1,$$

which leads to

$$|\omega| \leq 1$$

for any choice of $-1 \leq \cos g\pi \leq 1$.

These arguments provide evidence that the Du Fort and Frankel scheme is stable for realistic choices of r, σ and T.

5.5 Conclusion

We investigated three schemes that share two themes found in the Du Fort and Frankel finite difference scheme. The first theme is the improvement of accuracy by approximating the temporal derivative by making use of a two-sided approach. This approach is utilized by the Richardson scheme, and this adjustment leads to unconditional instability.

The second theme pertains to the conditional stability associated with the explicit scheme. By adjusting the approximation for the diffusion term in the partial differential equation, one obtains a reduction of mass along the main diagonal of the right hand side matrix in the matrix equation

$$\mathbf{M_L f^{i+1}} = \mathbf{M_R f^i} + \tilde{\mathbf{b}}.$$

This theme is employed in the MADE scheme leading to improved stability in the MADE scheme and is also responsible for stability associated with the Du Fort and Frankel scheme.

The principal compromise for stability is that the scheme is only conditionally consistent with the partial differential equation. This is a result of having error terms associated with the diffusion term of the partial differential equations of the form

$$T_{\text{Diffusion Term}} = X + Y\frac{k^x}{h^y}.$$

These schemes' consistency depends on the relative tempos at which $k, h \to 0$, i.e. the relative scales of x and y. The inconsistency of the MADE scheme seems more severe due to (i) the order difference between $x = 1$ and $y = 2$, demanding a high

number of temporal steps in relation to spatial steps, and (ii) the Y coefficient in the case of the MADE scheme including terms of $\frac{\partial f}{\partial q}$, which also occurs in the original partial differential equation. Inconsistency of the Du Fort and Frankel scheme is far more accommodating by having x and y of similar order and also due to the fact that the Y coefficient is in terms of $\frac{\partial^2 f}{\partial q^2}$, which is generally a more manageable quantity.
The main qualification of the magnitude of inaccuracy associated with inconsistency in the Du Fort and Frankel scheme pertains to areas where $\frac{\partial^2 f}{\partial q^2}$ becomes large. Experimentation lead us to believe that such inaccuracies will predominantly occur near discontinuities in the function f. Discontinuous behavior is often a feature of financial problems and consequently this topic will be taken further in later chapters.
A summary of the properties of the three schemes is presented in Table 5.1.

Property	**MADE**	**Richardson**	**Du Fort and Frankel**
Truncation Error	$\mathcal{O}\left(k + \frac{k}{h^2} + h^2\right)$	$\mathcal{O}(k^2 + h^2)$	$\mathcal{O}\left(k^2 + \frac{k^2}{h^2} + h^2\right)$
Consistency	$M > N^2$	Consistent	$M > N$
Stability	Practically	Unstable	Stable

Table 5.1: A summary of the properties of the MADE, Richardson and Du Fort and Frankel finite difference schemes.

Chapter 6

Miscellaneous topics: Convexity dominance and consistency improvement

6.1 Introduction

Spurious oscillations associated with convexity dominant partial differential equations is a well known problem [see for instance Seydel, 2004; Duffy, 2004b]. According to Duffy [2004b] these oscillations occur with schemes that makes use centered differencing in space combined with averaging in time (such as the Crank–Nicolson scheme) and when drift is high compared to diffusion. Seydel [2004] on the other hand describes oscillatory problems in terms of the Péclet number, which is defined as

$$\mathbf{Pe} = \frac{2r}{\sigma^2}\frac{h}{S}.$$

It is interesting to note the similarities of the Péclet number and the stability conditions of the MADE scheme (Section 5.2.3). Spurious oscillations is associated with a high Péclet number, which in turn makes instability in the MADE scheme more likely. Similarly, in the case of the heat equation, where convection is absent, $\mathbf{Pe} = 0$ and the MADE scheme becomes unconditionally stable.

The Crank–Nicolson scheme is often criticized for its susceptibility to spurious oscillations [Duffy, 2004b]. Using that scheme as a benchmark, we investigate the behavior

of the Du Fort and Frankel scheme in the presence of a very high Péclet number in order to determine if it is equally prone to such oscillations.

6.2 Convection dominated spurious oscillations

The numerical solution of convection dominated partial differential equations are known to produce spurious oscillations in the first derivative of the spatial variable[Duffy, 2004b; Seydel, 2004]. These oscillations scale with the Péclet number, which is defined as

$$\mathbf{Pe} = \frac{2r}{\sigma^2}\frac{h}{S}.$$

Duffy [2004a] argues that oscillatory behavior occurs when $A < 0$ (see equation 3.17). This places the restriction on h being

$$h \leq |\frac{2\alpha}{\beta}|,$$

which is similar to the definition of the Péclet number. In the extreme case, when $\sigma = 0$ the difference scheme approximates the hyperbolic equation

$$-\frac{\partial f}{\partial q} + \beta\frac{\partial f}{\partial s} + \gamma f = 0.$$

Initial errors are not dissipated leading to oscillations.
According to Duffy [2004b] the time averaging associated with the Crank–Nicolson scheme makes it especially prone to producing oscillations, while the fully implicit scheme is free of such behavior. Our own investigations do not support this notion, as is clearly visible in figure (6.1).

The delta of a European option is numerically calculated by making use of central differences, i.e.

$$\frac{\partial f}{\partial s} \approx \frac{f(s+h) - f(s-h)}{2h}.$$

The left hand side depicts the case where $\mathbf{Pe} = 1$ while the right hand side depicts $\mathbf{Pe} = 30$. Clearly the delta on the right hand side displays oscillatory behavior, unrelated to the finite difference scheme employed.

6.2.1 One sided convection differencing

Spurious oscillations associated with convection dominant equations are ascribed to the central differencing of the convection term [Duffy, 2004b]. We experiment with

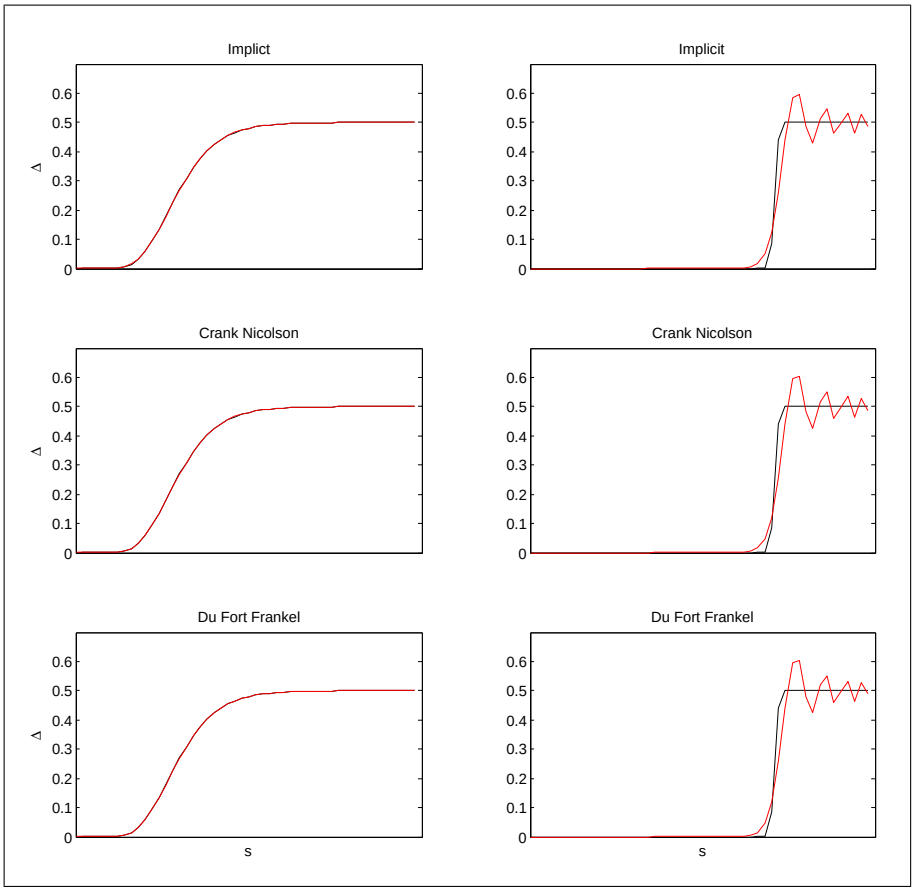

Figure 6.1: Oscillations associated with convection domination. The left hand side graphics show a numerical approximation of the delta of an European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$, while the right hand graphic shows the delta with $\sigma = 0.01$.

a different estimation, namely a one sided differencing approximation [Tavella and Randall, 2000, for instance]. We find the approximations by subtracting the Taylor expansion for f^i_{j+2} from Taylor expansion of $4f^i_{j+1}$ (and similarly by subtracting $4f^i_{j-1}$ from f^i_{j-2}).

$$\begin{aligned}\frac{\partial f}{\partial s} &\approx \frac{1}{2h}(-f^i_{j+2}+4f^i_{j+1}-3f^i_j)+\frac{1}{3}h^2\frac{\partial^3 f}{\partial s^3}-\frac{1}{4}h^4\frac{\partial^4}{\partial s^4}+\ldots, \text{ or}\\ \frac{\partial f}{\partial s} &\approx \frac{1}{2h}(3f^i_j-4f^i_{j-1}+f^i_{j-2})+\frac{1}{3}h^2\frac{\partial^3 f}{\partial s^3}-\frac{1}{4}h^4\frac{\partial^4}{\partial s^4}+\ldots.\end{aligned}$$

Substituting these approximation into the Black and Scholes partial differential equation leads to the Du Fort and Frankel difference equations

$$f^{i+1}_j = \begin{cases} \frac{-\beta kh}{h^2+2\alpha k}f^i_{j+2}+\frac{2\alpha k+4\beta kh}{h^2+2\alpha k}f^i_{j+1}+\frac{2\gamma kh^2-3\beta kh}{h^2+2\alpha k}f^i_j+\ldots & \\ \qquad \frac{2\alpha k}{h^2+2\alpha k}f^i_{j-1}+\frac{h^2-2\alpha k}{h^2+2\alpha k}f^{i-1}_j & \text{if } j\geq 3,\\ \frac{2\alpha k}{h^2+2\alpha k}f^i_{j+1}+\frac{3\beta kh+2\gamma kh^2}{h^2+2\alpha k}f^i_j+\frac{2\alpha k-4\beta kh}{h^2+2\alpha k}f^i_{j-1}+\ldots & \\ \qquad \frac{\beta kh}{h^2+2\alpha k}f^i_{j-2}+\frac{h^2-2\alpha k}{h^2+2\alpha k}f^{i-1}_j & \text{if } j\leq N-1.\end{cases}$$

Results obtained from experimentation lead to the conclusion that one sided differencing has little impact on the reduction of spurious oscillations. The algorithm (see A.2.4) alternates between upwards and downwards differencing.

6.3 Consistency improvements of the Du Fort and Frankel scheme

The inconsistency associated with the Du Fort and Frankel scheme originates from error terms of the form

$$\frac{k^x}{h^y}Z.$$

The relationship between x and y determines the severity of the inconsistency of the scheme, for instance the MADE scheme has $x=1$ and $y=2$ while the Du Fort and Frankel scheme has $x=2$ and $y=2$, which lead to more easily obtainable consistency. Figure 6.3 shows errors made with the Du Fort and Frankel scheme (left) compared to errors made with the Crank and Nicolson scheme (right). The figures at the bottom shows a similar error for both schemes obtained with a grid with $M=50$ and $N=20$. The figures at the top shows the Du Fort and Frankel scheme to be clearly inferior to the Crank and Nicolson scheme with $M=20$ and $N=50$.

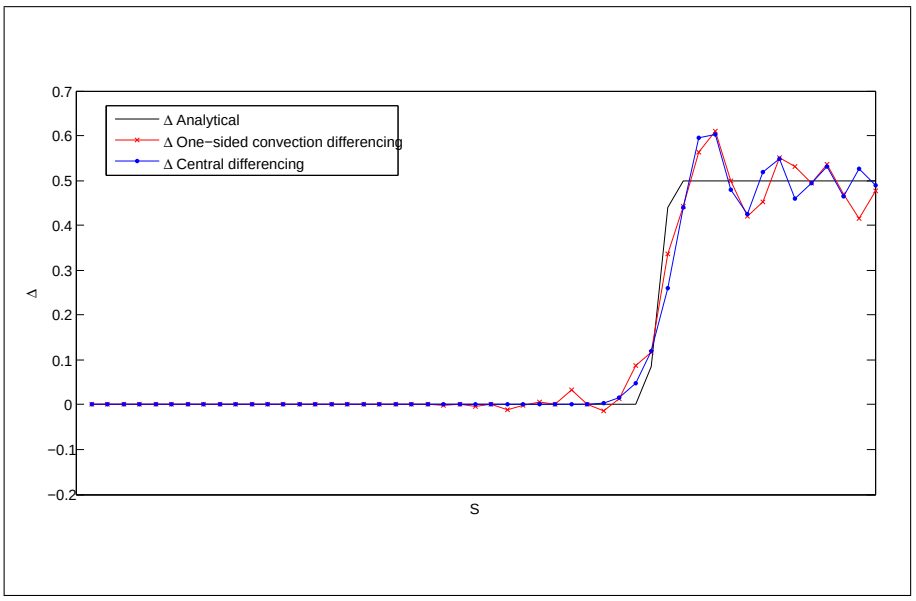

Figure 6.2: Oscillations created by two methods of convection approximation. Similar input parameters were used for both the Du Fort and Frankel scheme with central convection differencing and the Du Fort and Frankel scheme with one-sided convection estimation. The figure shows a numerical approximation of the delta of an European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.01$.

We experiment with a number of measures which could achieve satisfactory consistency for the Du Fort and Frankel scheme. The first measure is to simply apply a different mesh size such that errors for the Du Fort and Frankel scheme is comparable to that of the Crank and Nicolson scheme. A second measure is to employ Richardson's extrapolation in order to cancel terms of the order k^2/h^2.

6.3.1 Changing the mesh size

Since the Du Fort and Frankel scheme is explicit, computation is less taxing than for a similar implicit scheme such as the Crank and Nicolson scheme, which involves matrix inversion or an equivalent technique. Table (6.1) shows the time taken (in seconds) to compute different grid sizes by using the Crank–Nicolson scheme (see section (A.2.1) for the algorithm used) and the Du Fort and Frankel scheme (A.2.3). Even for relative small grid sizes, the Du Fort and Frankel scheme is far more efficient.

It is thus possible to increase the number of temporal steps such as to enforce a more satisfactory consistency for the Du Fort and Frankel scheme, and at the same time still achieve comparable or better results than the Crank–Nicolson scheme utilizing similar

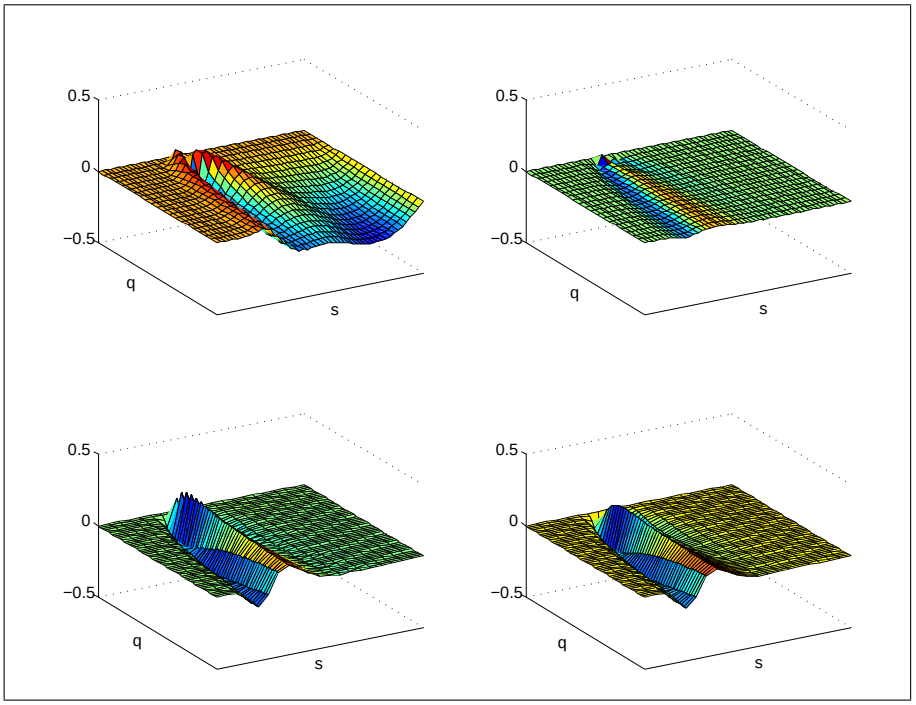

Figure 6.3: Errors associated with Du Fort and Frankel approximations (left) with $M = 20$ and $N = 50$ (top) and $M = 50$ and $N = 20$ (bottom) compared to errors associated with a Crank and Nicolson approximation (right) with similar grid settings top and bottom for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$.

Grid size	Du Fort Frankel	Crank–Nicolson	Factor
1000×1000	2.07	20.31	9.8
500×500	0.27	2.08	7.8
250×250	0.05	0.25	5.4

Table 6.1: Time to compute different grid sizes by using the Du Fort and Frankel scheme and the Crank and Nicolson scheme

resources[1]. Figure (6.4) depicts three scenarios. The top graphic is the error made with a $N = 250$ and $M = 250$ grid calculated with the Crank–Nicolson scheme. The middle graphic depicts a similar mesh calculated with the Du Fort and Frankel scheme. For similar grid sizes, a lower accuracy was achieved by the Du Fort and Frankel scheme, however the Du Fort and Frankel scheme only took 0.045 seconds to compute compared to 0.25 seconds for the Crank–Nicolson scheme. By setting a finer grid ($M = 1500$ and $N = 250$) the Du Fort and Frankel scheme achieved a similar accuracy than the Crank and Nicolson scheme and the time taken to compute was also similar, namely 0.23 seconds.

6.3.2 Consistency improvement by Richardson's extrapolation

Richardson's extrapolation [see for instance Feldman; Wilmott, 2000b] is a well known technique to improve the accuracy of numerical approximations. We experiment with the applicability of Richardson's extrapolation in order to improve the consistency characteristics of the Du Fort and Frankel scheme. From section (5.4.3) a solution for the Black–Scholes partial differential equation may be interpreted as

$$F_j^i = \hat{f}_j^i(k,h) + T_j^i,$$

where F_j^i is the exact solution of the partial differential equation and $\hat{f}_j^i(k,h)$ is the finite difference solution of F_j^i given a mesh with spatial step size h and temporal step size k. The truncation error is of the form

$$T_j^i \equiv k^2\epsilon_1 + h^2\epsilon_2 + \frac{k^2}{h^2}\epsilon_3 + \mathcal{O}(k^4) + \mathcal{O}(h^3) + \mathcal{O}(\frac{k^4}{h^2}).$$

By substituting T_j^i with its extended form, and discarding higher order terms, we write $\hat{F}_j^i$ as an improved approximation to F_j^i.

$$\hat{F}_j^i = \hat{f}_j^i(k,h) + k^2\epsilon_1 + h^2\epsilon_2 + \frac{k^2}{h^2}\epsilon_3.$$

We calculate 2 finite difference solutions, namely $\hat{f}_j^i(k_1,h)$ and $\hat{f}_j^i(k_2,h)$. By subtracting $\frac{k_1^2}{k_2^2}$ of the second solution from the first, we obtain

$$\left(\frac{k_2^2 - k_1^2}{k_2^2}\right)\hat{F}_j^i \quad = \quad \hat{f}_j^i(k_1,h) - \frac{k_1^2}{k_2^2}\hat{f}_j^i(k_2,h) + h^2\epsilon_2\left(\frac{k_2^2 - k_1^2}{k_2^2}\right),$$

[1] We measure resources here in terms of the time used to compute a result.

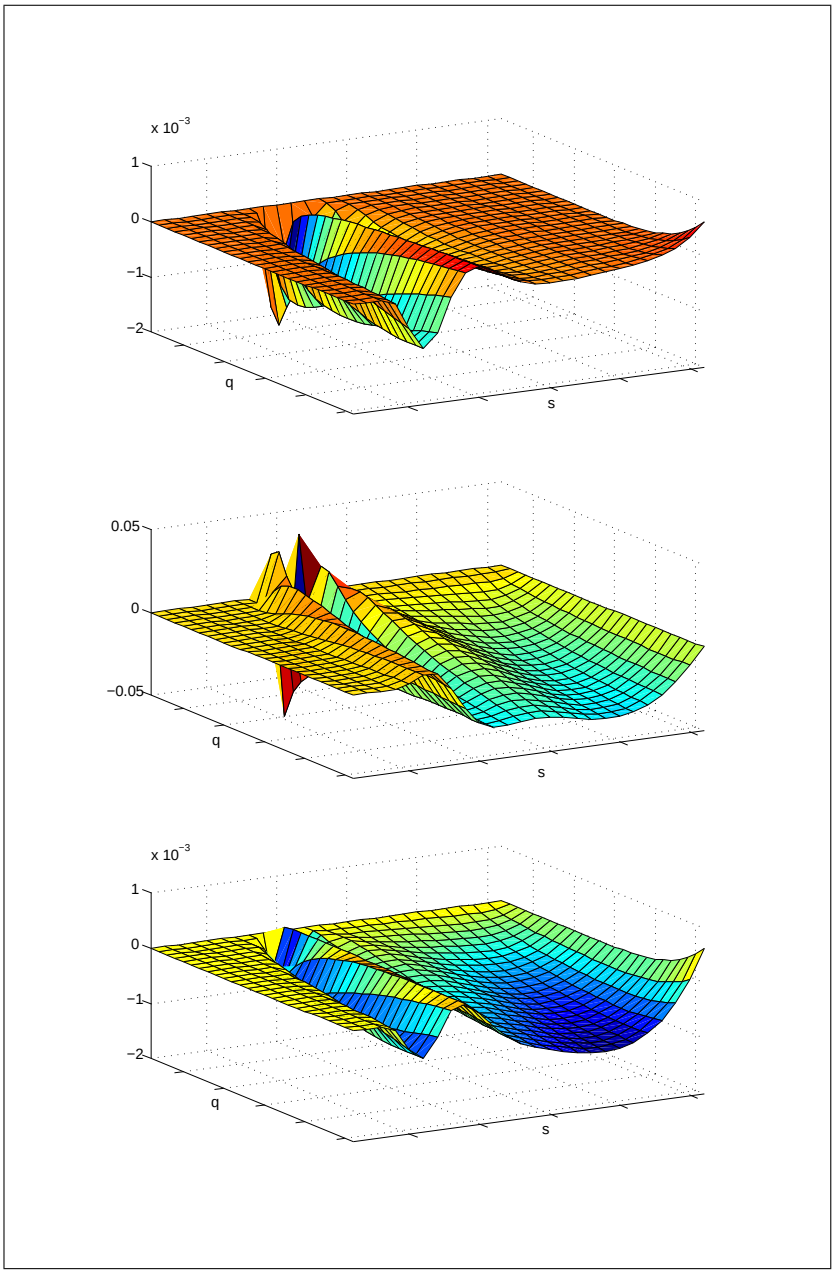

Figure 6.4: Errors associated with Crank and Nicolson (top) for an $M = 250$ and $N = 250$ grid compared to errors associated with Du Fort and Frankel approximations (middle and bottom). The middle figure is for a grid size of $M = 250$ and $N = 250$ while the bottom figure is for a grid size of $M = 1500$ and $N = 250$. The Du Fort and Frankel solution with the finer grid took similar computing resources than the Crank and Nicolson solution (0.23 seconds and 0.25 seconds respectively). The option that was computed was a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$.

which after rearranging terms give

$$\hat{F}_j^i = \frac{k_2^2 \hat{f}_j^i(k_1, h) - k_1^2 \hat{f}_j^i(k_2, h)}{k_2^2 - k_1^2} + h^2 \epsilon_2. \tag{6.1}$$

If we use $\hat{F}_j^i$ as our solution instead of either $\hat{f}_j^i(k_1, h)$ or $\hat{f}_j^i(k_2, h)$, we obtain a consolidated scheme with truncation error

$$T_j^i = \mathcal{O}(k^4) + \mathcal{O}(h^2) + \mathcal{O}(\frac{k^4}{h^2}).$$

These convergence properties are firstly superior to that of Crank and Nicolson, and secondly, its consistency properties is vastly superior to the unaltered Du Fort and Frankel scheme since the lowest order mixed error term is now $\mathcal{O}(\frac{k^4}{h^2})$ instead of $\mathcal{O}(\frac{k^2}{h^2})$. Figure (6.5) depicts two mesh sizes (top and middle) and a Richardson's extrapolation combining the two (bottom). The Richardson's extrapolation clearly produces superior results.

Even though a solution is required to be calculated twice, the total computational resources utilized is far superior to that of similar implicit schemes such as the Crank and Nicolson scheme. As a measure of comparison the above example took 0.004 seconds for the first solution and 0.002 for the second. The combined time taken by the Richardson's extrapolation was 0.0066 seconds. A similar error profile was obtained with the Crank and Nicolson scheme by using $N = 80$ and $M = 80$, taking 0.0169 seconds, 2.55 times longer than the Du Fort and Frankel scheme with Richardson's extrapolation. This is depicted in Figure (6.6). The top part depicts a Du Fort and Frankel scheme with $M = 180$. Although it utilizes similar computational resources than the Richardson's extrapolation (middle) with $M_1 = 120$ and $M_2 = 60$, its performance is clearly inferior compared to the Richardson's extrapolated Du Fort and Frankel scheme. A similar error surface was obtained with a Crank and Nicolson scheme (bottom) with $M = 80$ and $N = 80$, but since matrix inversion is involved with this scheme, it still took more than twice the time to compute than the Du Fort and Frankel scheme with Richardson's extrapolation. Comparing computing time with computing time, the Du Fort and Frankel scheme with Richardson's extrapolation vastly outperforms the Crank and Nicolson scheme!

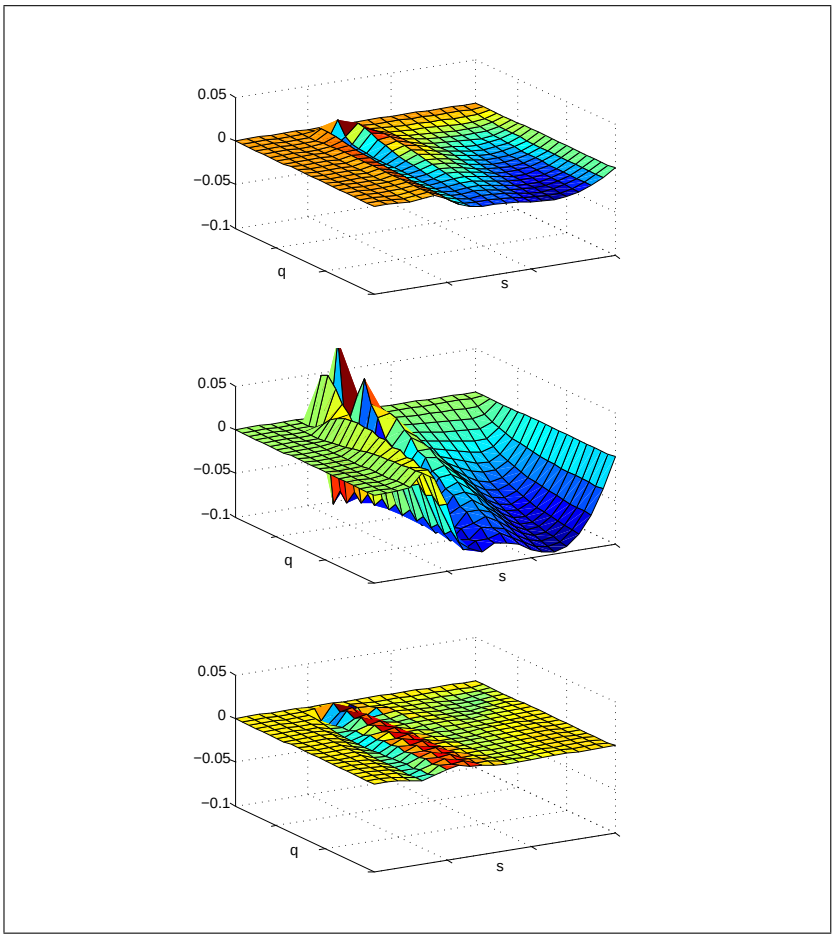

Figure 6.5: Errors associated with the Du Fort and Frankel scheme for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$. The number of spatial steps were $N = 100$ with $M_1 = 120$ (top), $M_2 = 60$ (middle), and a Richardson's extrapolation with $k_1 = T/M_1$ and $k_2 = T/M_2$ (bottom).

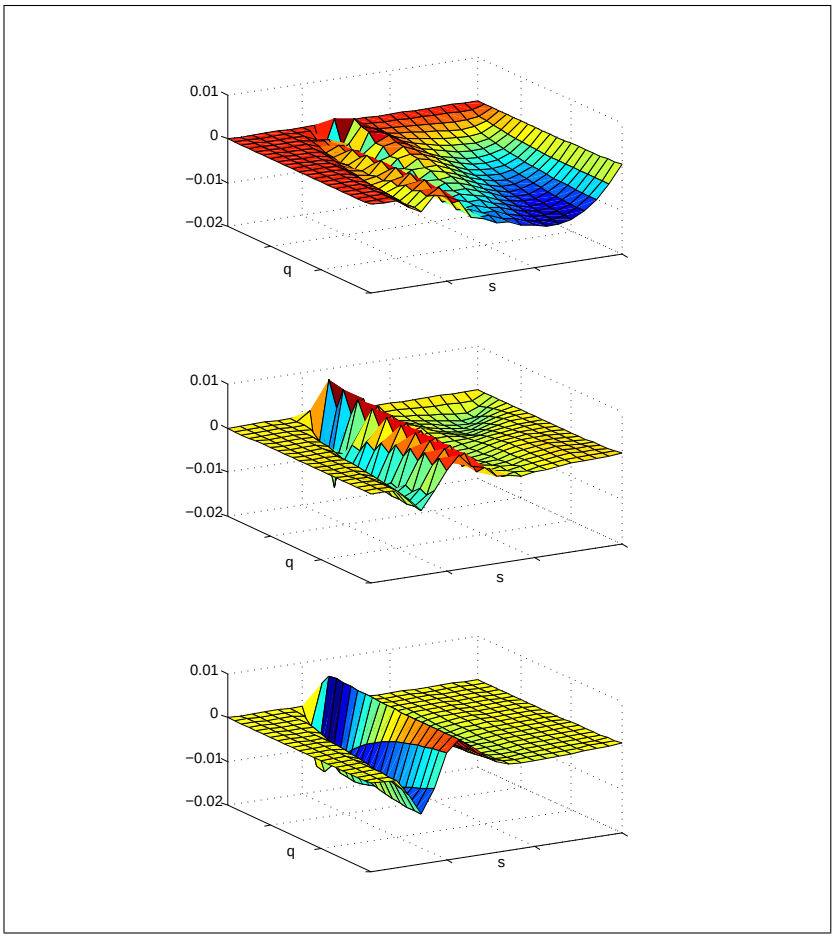

Figure 6.6: Errors for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$. The number of spatial steps were $N = 100$ with a Du Fort And Frank scheme with $M = 180$ (top), a Richardson's extrapolated Du Fort and Frankel scheme with $M_1 = 120$ and $M_2 = 60$ (middle), and a Crank and Nicolson scheme with $M = 80$ and $N = 80$ (bottom).

6.3.3 Comparison of techniques to improve consistency characteristics

Table 6.2 provides a qualitative comparison of the two measures to improve the consistency characteristics of the Du Fort and Frankel scheme.

Benchmark	**Change mesh size**	**Richardsons extrapolation**
Elimination of the problem	Gradual improvement	Marked improvement
Ease of implementation	Trivial	Minor algorithmic adjustment
Spin offs	None	Better truncation error
Time to achieve results comparable to Crank & Nicolson	Similar	Marked improvement

Table 6.2: A summary of the measures taken to improve consistency.

6.4 Conclusion

We investigated techniques to reduce oscillations associated with convexity dominance as well as techniques to reduce the impact of the inconsistency of the Du Fort and Frankel scheme with the Black and Scholes partial differential equation.

Some literature claim that oscillations associated with convexity dominance is a result of firstly the way in which the temporal derivative is discretised by averaging, such as is the case with the Crank and Nicolson scheme, and secondly as a result of the central convection differencing. Neither these claims were consistent with our experiments. Firstly, both the fully implicit scheme (that makes use of a one-sided time discretisation) and the Du Fort and Frankel scheme (that makes use of a central time discretisation) display oscillatory behavior in the presence of convexity dominance. Secondly, by using one-sided convexity discretisation also did not improve the magnitude of oscillations. We concur with the conclusion of Seydel [2004] that oscillatory behavior is an inherent characteristic of the model and not the scheme. Furthermore, for most realistic problems in finance, these oscillations have minimal impact on the solution. The impact of oscillations on the "Greeks" were not exhaustively tested.

Insofar the inconsistency of the Du Fort and Frankel scheme goes, we experimented with two ideas. The first was to simply increase the number of temporal steps, on the

premise that explicit schemes such as the Du Fort and Frankel scheme utilizes less computational resources than implicit schemes. Implicit schemes require matrix inversion which is relatively complex compared to the computations involved with explicit schemes. The implication is that explicit schemes with an increased number of temporal steps are computationally comparable to implicit schemes with less temporal steps. A second idea that was explored was to make use of Richardson's extrapolation in order to cancel inconsistent error terms. The advantage of this technique is that not only does the scheme become more consistent, it also converges quicker. Our experimentation leads us to the conclusion that the Du Fort and Frankel scheme with Richardson's extrapolation vastly outperforms the Crank and Nicolson scheme.

Chapter 7

Part I conclusion and summary

In Part I we established – via theoretical analysis – a context, which provide us the means to compare different finite difference schemes to one another. The objective measures were local truncation error, consistency and stability. Often in literature these three measures provide sufficient evidence that one scheme ought to be chosen above another. It thus comprises the full suite of arguments used in order to discriminate between schemes.

We augment this suite of discriminating factors with the more subtle property of computational effort and ease of implementation. Explicit schemes are often accused of utilizing more time steps in order to stabilize. This accusation rarely offers the qualification that explicit schemes can often afford extra temporal steps since they are computationally less taxing than their implicit counterparts. Our prime discriminating factor is one where we measure computational effort in order to achieve similar results. This will also be the prime determining factor in the second part of this document.

Our system of analysis firstly defined the problem. We investigated numerical techniques in order to compute the arbitrage free price of contingent claims. In Chapter 2 we delved into the works of Black and Scholes [1973] and Merton [1973], amongst others, in order to describe the price of a contingent claim as a partial differential equation, known as the Black and Scholes equation. The Black and Scholes equation describes the problem, and the remainder of Part I was devoted to describe and analyse the intended method to solve the problem, namely the finite difference method. Chapter 3 provides the general theoretical description of finite difference techniques.

The reader is introduced to the construction of a finite difference mesh, specification of boundary conditions and estimation of derivatives. We evaluate the problem from two perspectives namely an initial value partial differential Black and Scholes equation, and a simplified heat equation. Three finite difference techniques are introduced namely the fully explicit scheme, the fully implicit scheme and the Crank and Nicolson scheme. These three schemes are then shown to be distinct cases of a more general scheme, namely the Theta scheme. By changing the value of a single parameter, the Theta scheme becomes any of the fore mentioned schemes. Douglas schemes employ a more optimal choice for the Theta parameter, improving the local truncation error in the case of the heat equation, but not for the Black and Scholes partial differential equation.

The three main discriminating factors of finite difference schemes, local truncation error, consistency and stability are discussed in Chapter 4. It is shown that the fully explicit scheme is only conditionally consistent with the partial differential equation. Testing stability for the Black and Scholes partial differential equation is more complex than conducting similar tests on the heat equation. Although our methodology is similar to Wilmott [2000b], we obtain different stability conditions.

Chapter 4 concludes the general section of finite difference theory. The remainder of the section expanded on themes in order to arrive at the Du Fort and Frankel scheme. Chapter 5 considers the main themes that differentiates the Du Fort and Frankel scheme from the classical Theta schemes namely second order local truncation error for explicit schemes and unconditional stability. Two other schemes that share aspects of these themes were analysed. The MADE scheme shares unconditional stability. This is achieved by sacrificing unconditional consistency. The Richardson scheme shares the theme of a second order accurate truncation error, but suffers from unconditional instability as a result of the required adjustment to achieve the better local truncation error. The Du Fort and Frankel scheme makes use of techniques present in both these schemes in order to obtain second order accuracy and simultaneously achieve unconditional stability. Like the MADE scheme (only less severe) it is only conditionally consistent with the partial differential equation.

We finally investigate the Achilles heel of the Du Fort and Frankel scheme namely conditional consistency. It was shown that at least two techniques effectively reduce the problem to a manageable quantity. These techniques are to increase the number of time

steps or to make use of Richardson's extrapolation in order to obtain better consistency characteristics.

In addition to the analysis of the consistency properties of the Du Fort and Frankel scheme, we also analysed the impact of convexity dominance, which is a potential characteristic of the Black and Scholes equation. It was shown that although spurious oscillations are possible with the Du Fort and Frankel scheme, this was a result of the presence of convection terms in the partial differential equation, rather than a property of the scheme. We failed to replicate results of Duffy [2004b], implicating inherent characteristics of schemes, namely central convection differencing combined with averaging in the temporal direction, as being the culprits causing spurious oscillations.

The concluding observation from Part I is that the Du Fort and Frankel scheme exhibits promising characteristics which may prove valuable in the pricing of derivative securities. These are:

- It is explicit, and thus computationally more efficient than implicit schemes.
- It is unconditionally stable.
- It is second order accurate in both the temporal and spatial dimension.
- It is only conditionally consistent with the partial differential equation, but this can effectively be managed. By making use of Richardsons extrapolation, we not only managed to reduce the effect of inconsistency considerably, we also succeeded to obtain additional accuracy.

The foundation was thus laid for the next phase of scrutiny, namely to expose the Du Fort and Frankel scheme to a number of challenging pricing conundrums that frequently occur in finance.

Part II

Recurring numerical problems in finance

Introduction to Part II

In part II we investigate the ability of Du Fort and Frankel finite difference scheme to overcome numerical difficulties that arise from financial problems. These problems often challenge the assumptions underlying the derivation of the of the partial differential equation [Black and Scholes, 1973; Merton, 1973] and often requires adaptation of the finite difference scheme in order to solve the newly formulated problem [Wilmott, 2000b]. The changes that one scheme undergo are not necessarily identical to those required for other schemes with the most profound differences between implicit and explicit schemes.

We argue a case for explicit schemes on the basis of reduced computational effort. This approach was amongst others advocated by Hull and White [1990]. The counter argument is often based on the premise that explicit schemes are conditionally stable, or alternatively, only conditionally consistent. The Du Fort and Frankel scheme suffers from occasional inconsistency [Smith, 1984] which is debatably worse than instability under certain conditions as its impact is subtle and may go unnoticed if we do not have the benefit of comparing results to known solutions.

Part II is structured in the following way:

- In Chapter 8 we investigate the impact of jumps and especially dividends on the formulation of the problem (the Black and Scholes partial differential equation) and also the impact it has on the implementation of finite difference schemes. We establish whether the Du Fort and Frankel scheme requires alternative adaptations to other schemes.

- Chapter 9 scrutinizes the numerical difficulty that results from singularities and steep gradients that is often a feature of problems pertaining to finance. We investigate solutions that are regularly used by other schemes and also derive a unique solution for the Du Fort and Frankel scheme, namely a third order interpolated mesh refinement.

- In Chapter 10 we investigate the problem of free boundary values and proceed to compare the techniques to solve such problems. These problems are frequently occurring in finance and may be one of the driving motivations in using finite difference schemes. We investigate techniques to solve these for both implicit

and explicit schemes and compare results obtained by both with other numerical results.

- We extend the one dimensional case studied so far to a multidimensional case in Chapter 11 and use a two-dimensional problem to derive the difference equation for the Du Fort and Frankel scheme. The problem is chosen such that an analytical solution exists and results obtained from the Du Fort and Frankel scheme are compared to the analytical solution. We experiment with a boundary free scheme in order to reduce the total number of grid nodes that require computation.

These topics represent a small section of the universe of numerical difficulties that arise from financial problems. We chose these due to their frequent appearance on the financial landscape.

Chapter 8

Jumps and dividends

8.1 Introduction

Modern theory concerning the arbitrage free pricing of contingent claims stems from works published by Black and Scholes [1973] and Merton [1973]. Black and Scholes explicitly assumes an underlying security that pays no dividends or other distributions. Merton derives a partial differential equation for warrant prices in the presence of dividends, but adds to his commentary that such partial differential equation has no simple solution. The Merton case for continuous dividends is often incorporated as a special case in a generalized Black-Scholes formula [Haug, 1998] which has become the analytical norm for continuous dividends.

Pricing contingent claims by assuming continuously paying dividends is not an ideal compromise. Equities pay dividends sporadically at discrete time intervals [Haug et al., 2003]. Such dividend payments impact on the stock price process; consequently on the price of the underlying stock and ultimately on the value of the contingent claim.

In the Black-Scholes and Merton world, options are evaluated from an underlying security that has a continuous stochastic process (equation (2.1)). Evaluating an option at maturity time T assumes a continuous price path for the share over the entire period $[t_0, T]$. If a dividend was to be paid during this interval, a discontinuous jump would have resulted [Björk, 2004], impacting on the terminal distribution share prices, and consequently the value of a contingent claim depending on the terminal price of the share.

Dividend payments are unlike other price jumps because it is subject to an arbitrage argument [Björk, 2004] that enables accurate prediction of the dividend impact on the share price. This argument goes further and relates the jump to the value of any contingent claim on the share.

Since dividend payments present jumps in the price of the underlying instrument at discrete time intervals, claims based on dividend paying equities must be regarded as path-dependent, and thus numerical techniques like the finite difference method present a practical means to price such claims.

We model discrete dividends in two distinct ways that may affect the procedure employed. Dividend payments are often presented in the form $D : \mathbb{R} \to \mathbb{R}$. This function usually takes one of two forms namely *fixed* discrete dividends and *fractional* discrete dividends. These two methodologies can be presented as

$$\delta(S(t)) = \alpha\beta(t),$$

where α is constant and

$$\beta(t) = \begin{cases} 1 & \text{if a dividend is paid at time } t \\ 0 & \text{if a dividend is not paid at time } t \end{cases}$$

in the case of *fixed* dividends, and

$$\delta(S(t)) = \alpha\beta(t)S(t),$$

in the case of *fractional* dividends.

The main distinction between fixed and fractional dividends is that the latter is a stochastic function, depending on on the underlying price. This geometric configuration simplifies matters somewhat, compared to deterministic characteristics of fixed dividends. In literature the term “discrete” is often reserved for fixed dividends [Haug et al., 2003, for instance] because the geometric nature of fractional dividends allow for analytic solutions for contingent claims where the payoff function depends on the terminal value of the underlying instrument. However, when the payoff also depends on the path of the underlying instrument, exact analytical solutions do not exist, and for this reason we also consider fractional dividends as “discrete”.

8.1.1 A theoretical framework for dividends

Assume that over the life of the contingent claim $[t_0, T]$ there are n dividends payable. Let τ be a $1 \times n + 2$ vector of deterministic times at which dividends are paid.

$$\tau = \{0, T_n, T_{n-1}, \ldots, T_2, T_1, T\}, \tag{8.1}$$

where $0 < T_n < T_{n-1} < \ldots < T_2 < T_1 < T$. During the periods between dividends, i.e. during the intervals $[0, T_n)$, $[T_n, T_{n-1})$, ..., $[T_2, T_1)$ and $[T_1, T]$, the underlying share follows the process described by equation 2.1. Note that all the intervals except the last one are half-open. Each period starts "just after" a dividend has been paid and ends "just before" the next dividend. If a dividend is paid at precisely time t, then we denote the time "just before" that event by $t- \equiv t - dt$.

Assume that discrete dividends are functions of the form

$$\delta = \delta(S(t-)).$$

The dividend amount is thus known "just before" the dividend is paid. As noted before this function can take one of two forms, either δ is *fixed* in the case of a deterministic Rand dividend or δ is a *fractional* dividend when it is a function of $S(t-)$.

8.1.2 Dividends and jumps

The Black–Scholes partial differential equation was derived for an equity that follows geometric Brownian motion with no jumps. Black-Scholes analysis assumes continuous trading and the absence of dividends. Merton [1973] relaxes the dividend assumption and in a later paper [Merton, 1976] also introduces jumps in the process in order to do away with the continuous trading assumption. It was shown that the introduction of discontinuous jumps cause higher option prices and that options prices on such shares cannot be obtained by means of arbitrage pricing. The argument is based on the premise that these jumps cannot be anticipated and that they are random on both the spatial and temporal axes. In this respect jumps due to dividend payments are different. Dividends can at least in the short term be anticipated with reasonable certainty. Björk [2004] describes an arbitrage argument that states that the jump during the infinitesimal progression in time from cum-dividend to ex-dividend can only be the value of the

dividend[1]. Between dividend payments, the stock price dynamics is left intact.

Over dividend payments, we have the following fundamental relationship:

$$F(T_k-, S(T_k-)) = F(T_k, S(T_k-) - \delta(S(T_k-))), \tag{8.2}$$

where $k = 1, 2, \ldots, n-1, n$ and F is the price of a contingent claim.

Björk suggests an argument for dividend payments, which involves the solving of an iterative processes. During the interval $[T_1, T]$ there are no dividends. We may thus proceed to calculate the value of a contingent claim at time T_1. The value for $F(T_1)$ is thus known. By equation (8.2), this is also the value for the contingent claim at time T_1-, and since the period $[T_2, T_1)$ is per definition free of dividends, we may proceed to calculate the value of the contingent claim at time T_2, which by equation (8.2) also equals the value for the claim at T_2-. We may thus proceed from dividend payment to dividend payment until we reach time period $t = 0$.

Using this argument to price options analytically involves nested integrals, which are problematic to solve.

The finite difference method is an effective means to overcome nested integrals by inherently integrating over the spatial domain at each time step. The advantage of the finite difference method is that it offers us a degree of freedom to manipulate the spatial domain at ex-dividend or cum-dividend dates [Wilmott, 2000b]. We are confronted with two classes of action that may be taken. The first is to make use of an escrowed dividend model. These models were developed to alter analytical models in order to approximate the presence of dividends. We are of the opinion that such methods are ineffective (and not arbitrage free [Haug et al., 2003, amongst others]) and that they do not utilize the versatility of the finite difference method to its full potential. We therefore only briefly discuss these methods.

A second class of action to be considered is to physically model dividends in the finite difference framework by interpolating the solution on to the spatial domain that coincides with cum-dividend prices.

[1] It is assumed that the dividend is paid on the ex-dividend event. In practise it is paid at a later time and one thus has to provide for a time value component which will be ignored in this analysis.

8.2 Fractional dividends.

Fractional dividends are less problematic to solve than fixed dividends. Merton [1973] has shown that continuous dividends still result in an analytical solvable partial differential equation. Since discrete fractional dividends share the geometric nature of the underlying process, we may attempt to replace a vector of discrete payments by a single dividend yield. For each time in τ we iteratively calculate a stock price and then by the arbitrage argument in equation (8.2) reduce the share price with the fractional dividend as follows:

$$\begin{aligned}
S(T_n) &= S(0)\exp\left((r-\frac{1}{2}\sigma^2)T_n+\sigma W_{T_n}\right)(1-\delta(S(T_n))) \\
S(T_{n-1}) &= S(T_n)\exp\left((r-\frac{1}{2}\sigma^2)(T_{n-1}-T_n)+\sigma(W_{T_{n-1}}-W_{T_n})\right) \\
&= S(0)\exp\left((r-\frac{1}{2}\sigma^2)T_{n-1}+\sigma W_{T_{n-1}}\right)((1-\delta(T_n))(1-\delta(T_{n-1}))) \\
&\vdots \\
S(T) &= S(0)\exp\left((r-\frac{1}{2}\sigma^2)T+\sigma W_T\right)\prod_{i=1}^{n}1-\delta(T_i). \qquad (8.3)
\end{aligned}$$

By setting equation 8.3 equal to the solution of equation 2.1, i.e.

$$S(T) = S(0)\exp\left((r-y-\frac{1}{2}\sigma^2)T+\sigma W_T)\right),$$

where y is the continuous dividend yield, we may solve for a dividend yield that will result in a terminal stock price distribution which is identical to the terminal stock price process resulting from the iterative process in equation (8.3)

$$S(0)\exp\left((r-\frac{1}{2}\sigma^2)T+W_T\right)\prod_{i=1}^{n}1-\delta(T_i) \quad = \quad S(0)\exp\left((r-y-\frac{1}{2}\sigma^2)T+W_T\right).$$

Rearranging terms lead to

$$y \quad = \quad -\frac{1}{T}\sum_{i=1}^{n}\log\left(1-\delta(T_i)\right). \qquad (8.4)$$

Using the calculated value of y from equation 8.4 results in stochastic process which has an identical *terminal* distribution to the real process with discrete dividends. Options which only depend on the terminal value of the spot price can therefore be priced in this way, but any form of path dependency will result in inaccuracies.

8.3 Escrowed dividends

Escrowed dividend models do not simulate the dynamics of the underlying security, instead the idea is to manipulate some of the input parameters of an analytic model in order manipulate option prices such that they appear correct for the dividend case. Although reasons for their use in analytical models are perfectly valid, such as time to compute, the fundamental reasoning remains incorrect. Often in literature escrowed dividends are presented as the correct procedure to account for dividends even with numeric techniques [Black, 1975; Hull, 2003, for instance]. This may be as a result of some additional complexity associated with implementing proper dividend techniques, but in general the finite difference method presents an excellent opportunity to correctly model dividends. We include the section on escrowed dividends for review purposes only.

8.3.1 Change in spot price only

One of the most frequently used methods to estimate options for shares that pays fixed dividends is to lower the current share price by the sum of the present values of the dividends that are expected until the maturity of the option. It is often contextualized in literature as the way in which options on shares with dividends ought be treated, amongst others by Black [1975], Hull [2003]; and often as a means to treat dividends numerically, for instance Clewlow and Strickland [1998].

This method is based on the premise that once a dividend is known in advance, then the share price dynamics consists of two parts [Chance et al., 2002], a stochastic part, $G(t)$ and a deterministic part consisting of the present value of known dividends,

$$D(t) \equiv \sum_{t \leq t_i \leq T} d_i e^{-r(t_i - t)},$$

where d_i and t_i are denoting the discrete dividends and dividend dates respectively. The option price is then determined by using the stripped share price

$$G(t) = S(t) - D(t),$$

instead of $S(t)$ in analytical models.

Even though this method has merit – specifically it may be debated that this is an improvement on conventional Brownian motion in the sense that once dividends are

known they are deterministic by definition, it remains fraud in the Black-Scholes-Merton (BSM) world. The BSM model assumes that the entire stock price follows geometric Brownian motion (irrespective of whether this is the best way to model share prices...) and from this model option prices are computed. By assuming that only part of price has variance will result in too little variance for the entire process. Consider two stock price processes, $A(t)$ and $B(t)$. The first follows geometric Brownian motion between dividend payments, consistent with equation 8.2. Process $B(t)$ is assumed to consist of two parts $B_a(t)$, which is the share price stripped of its dividends, which follows geometric Brownian motion, and $B_b(t)$ which is the present value of the dividends during the period $t = [t_0, T]$. Without loss of generality we assume only a single dividend, $D(\tau)$ paid at time τ. Frisling [2002] shows that although both processes have identical expectations, the variance of their terminal distributions differ.

$$\begin{aligned} A(\tau) &= A(t_0)\exp[(r-\frac{1}{2}\sigma^2)(\tau-t_0)+\sigma(W(\tau)-W(t_0))]-D(\tau) \\ A(T) &= A(\tau)\exp[(r-\frac{1}{2}\sigma^2)(T-\tau)+\sigma(W(T)-W(\tau))] \\ &= A(t_0)\exp[(r-\frac{1}{2}\sigma^2)(T-t_0)+\sigma(W(T)-W(t_0))] \\ &\quad -D(\tau)\exp[(r-\frac{1}{2}\sigma^2)(T-\tau)+\sigma(W(T)-W(\tau))]. \end{aligned}$$

The process for $B(t)$ is given by

$$\begin{aligned} B(T) &= B_a(t_0)\exp[(r-\frac{1}{2}\sigma^2)(T-t_0)+\sigma(W(T)-W(t_0))] \\ &\quad -D\exp[r(T-\tau)]. \end{aligned}$$

The incorrect stock price distribution at maturity results in too low option prices. The mispricing worsens for dividend payment that occur later in the life of the option [Haug et al., 2003].

8.3.2 Spot and volatility adjustments: The Chriss model.

Chriss [1997] suggests an adjustment to both the spot price and volatility. The volatility adjustment $\tilde{\sigma}$ satisfies the relation

$$\tilde{\sigma}\tilde{S}_t = \sigma S_t.$$

The adjusted volatility is given as

$$\tilde{\sigma} = \frac{\sigma S_t}{S_t - \sum_{t\le t_i\le T} d_i e^{-r(t_i-t)}}. \tag{8.5}$$

The volatility adjustment corrects the total volatility error associated with the spot only adjustment model above, but may overestimate volatility especially for dividends that occur early in the life of the option [Haug et al., 2003].

8.3.3 The Haug & Haug and Beneder & Vorst approach.

Haug et al. [2003] discuss another approach independently discovered by Haug & Haug and Beneder and Vorst [2001]. This approach adjusts volatility in such a way as to make provision for the timing of dividends. Effectively it attempts to make volatility a function of time, but since closed form formulas for options do not accept multiple volatilities, it makes a timing-weighted average volatility. The time-weighted adjusted volatility is given by

$$\begin{aligned}\check{\sigma}^2 &= \left(\frac{\sigma S}{S-\sum_{i=1}^{n} S - d_i e^{rt_i}}\right)^2 (t_1 - t_0) + \\ &\quad \left(\frac{\sigma S}{S-\sum_{i=2}^{n} S - d_i e^{rt_i}}\right)^2 (t_2 - t_1) + \ldots + \sigma^2(T - t_n) \\ &= \sum_{j=1}^{n} \left(\frac{\sigma S}{S-\sum_{i=1}^{n} S - d_i e^{rt_i}}\right)^2 + \sigma^2(T - t_n). \qquad (8.6)\end{aligned}$$

Haug et al. [2003] remark that this adjustment, although it appears to produce better results than the ones mentioned earlier, is still without a sound theoretical base and may therefore be risky to trust.

8.3.4 Spot and strike price adjustments: The Bos & Vandermark approach.

Bos and Vandermark [2002] deviates from the school of thought where volatilities are adjusted to correct the spot price process for escrowed dividend models. Instead of making adjustments to the volatility of the stock price process they adjust the strike price of the option. Their reasoning is that if a dividend occurs very early in the life of the option, then using an adjusted spot price like in the escrowed dividend model produces very reliable option prices. However, when a dividend is paid late in the life of an option, just before expiration, then a better strategy to adopt is to raise the strike price.

Each dividend is divided into two parts, a near part ($X_n(T)$) and a far part ($X_f(T)$).

$$\begin{aligned} X_n(T) &= \sum_{i=1}^{n} \frac{T - t_i}{T} d_i e^{-rt_i}, \\ X_f(T) &= \sum_{i=1}^{n} \frac{t_i}{T} d_i e^{-rt_i}. \end{aligned} \tag{8.7}$$

The near part is subtracted from the initial spot price while the far part is added to the strike price. The ultimate effect is that the later a dividend occurs, the more it raises the strike and the less it lowers the initial spot price. The earlier a dividend occurs, the more it lowers the initial spot price and the less it raises the strike price.

8.3.5 Comparisons between escrowed dividend models

Table 8.1 summarizes observations relating to escrowed dividend models.

Model	**Accuracy with early dividend**	**Accuracy with late dividend**	**Comment**
Change in spot price only	Fair. Model is prone to underestimation.	Low accuracy	Not arbitrage free
Spot and volatility change	Low accuracy	Fair. Model is prone to overestimation.	Not arbitrage free
Beneder & Vorst, and Haug & Haug	Good	Good	No sound theoretical base
Bos & Vandermark	Good	Good	No sound theoretical base

Table 8.1: A comparison between escrowed dividend models.

8.4 Direct modeling of dividends in the finite difference framework

Wilmott [2000b], Oosterlee et al. [2004] and Leentvaar and Oosterlee [2006] suggests a procedure where option prices are interpolated on the spatial grid in order to accom-

modate discrete dividends. This technique is similar to one suggested by Schroeder [1988] to ensure node reconnection at the cum-dividend date when using binomial trees. Wilmott [2000b] argues that the interpolation technique must be at least of the same order of accuracy as the finite difference scheme, in this case $\mathcal{O}(h^2)$. A linear interpolation technique is suggested as its accuracy matches that of the scheme. Oosterlee et al. [2004] makes use of a 4^{th} order Lagrange interpolation as they employ a 4^{th} order scheme. We make use of a natural cubic spline interpolation, which is of $\mathcal{O}(h^3)$ accuracy. The Du Fort and Frankel scheme has a spatial local truncation error of $\mathcal{O}(h^2)$, which is lower than the natural cubic spline interpolation, consequently the interpolation technique does not impact on the accuracy of the overall scheme.

Two aspects of this procedure receives in our opinion too little attention in the afore mentioned literature. The first is that boundary conditions are required to be adjusted, and the second, applicable to schemes that involve multiple time steps such as the Du Fort and Frankel scheme, is that both cum dividend and ex dividend prices require interpolation. A typical algorithm to calculate an option with the Du Fort and Frankel scheme is shown:

- Given a dividend payable at time $\tau \equiv T - \nu k.$, compute vector $\mathbf{f^i}$ for $i = 1, 2, \ldots, \nu - 1, \nu$.
- At time τ, the vector $\mathbf{f}^\nu = (\hat{f}_1^\nu, \hat{f}_2^\nu, \ldots, \hat{f}_N^\nu, \hat{f}_{N+1}^\nu)$ maps to the spatial vector $\mathbf{s} = (s_\chi, s_\chi + h, \ldots, s_\psi - h, s_\psi)$. In order to proceed through time, we must transform the ex-dividend contingent claim prices to cum-dividend prices, by adhering to the equality given by equation (8.2). The solution vector $\mathbf{f}^\nu$ is thus required map to spatial vector

$$\mathbf{s}* = \begin{cases} \mathbf{s} + \delta(\tau) & \text{for fixed dividends, and} \\ \mathbf{s}(1 + \delta(\tau)) & \text{for fractional dividens.} \end{cases}$$

- Since vector $\mathbf{s}*$ does not coincide with real grid points, we need to interpolate the solution vector $\mathbf{f}^\nu$, which coincides with $\mathbf{s}*$ to coincide with with vector $\mathbf{s}$, which are points on the grid. By interpolation, the spatial vector $\mathbf{s}$ becomes a vector of cum-dividend underlying prices.
- This procedure is simple in a package such as Matlab. Listing (8.1) illustrates the interpolation procedure in Matlab.

```
tempvec = interp1(s_, V(:,i), s, 'spline');
V(2:N,i) = tempvec(2:N);
```

Listing 8.1: Matlab code showing option price interpolation to new underlying price vector

- The boundary conditions for times $t = [\tau, t_0]$ must be re-specified. We found the most accurate way in which to conduct this step is to find s_ψ from the boundary conditions, i.e for a European call option

$$\hat{f}^{\nu}_{N+1} = s_\psi + X \exp(-r(T - \tau)).$$

  Once the new s_ψ is known, the boundary values for times $t = [\tau + k, t_0]$ may be calculated.

- The interpolation enables the solution vector $\mathbf{f}^{\nu}$ at time τ to map to spatial vector $\mathbf{s}$. At time $\tau + k$, form the equation

$$\mathbf{f}^{\nu+1} = \mathbf{M}\mathbf{f}^{\nu} + \ddot{D}\mathbf{f}^{\nu-1} + \tilde{\mathbf{b}}^{\mathbf{i}},$$

  the Du Fort and Frankel scheme requires solutions vectors $\mathbf{f}^{\nu}$ and $\mathbf{f}^{\nu-1}$ in order to compute vector $\mathbf{f}^{\nu+1}$. It must observed that the spatial vector $\mathbf{s}$ at time $\tau + k$ comprises of cum-dividend prices, therefore the solution vector $\mathbf{f}^{\nu-1}$ maps to cum-dividend prices. In order to avoid incorrect and oscillatory behavior, solution vector $\mathbf{f}^{\nu-1}$ must also be interpolated in order to map to ex-dividend prices. This step differentiate the procedure for a multi time step scheme from the classical two step schemes such as the Crank and Nicolson scheme.

- This process repeats for each dividend.

Listing (8.2) shows the main iterations involved when the procedure of interpolating option prices to account for dividends is followed. The variable $Divs$ contains a list of dividends in the form $Divs = [Type, Time, Amount]$; where $Type$ can be either 'Fixed' or 'Fractional', $Time$ is the ex-dividend time measure from inception and $Amount$ is the dividend amount. A further variable $numDivs$ is declared to keep track of which dividends have been taken into account as we progress though time.

```
D_Vec = V(:,1);          %Two timestep vector
for i = 3:M+1
    V(2:N,i)= MatrixM*V(2:N,i-1) + ...
              MatrixBoundary(1:N-1,i-1) + ...
              D(2:N)'.*D_Vec(2:N);
        D_Vec = V(:,i-1);          %f(i-1) vector for the next time step
    if numDivs >= 1 %There are dividends
        if T-(i-1)*k >= Divs{numDivs,2} && T-i*k < Divs{numDivs,2}
         %Ex dividend date
         if strcmp(Divs{numDivs,1},'Fixed')
            s_ = s + Divs{numDivs,3};
            tempvec = interp1(s_, V(:,i), s, 'spline');
            newSmax = tempvec(N+1)+X*exp(r*(q(i)-T));
            V(N+1,i:M+1) = newSmax - X*exp(r*(q(i:M+1)-T));
          elseif strcmp(Divs{numDivs,1},'Fractional')
            s_ = s *(1+ Divs{numDivs,3});
            tempvec = interp1(s_, V(:,i), s, 'spline');
            newSmax = tempvec(N+1)+X*exp(r*(q(i)-T));
            V(N+1,i:M+1) = newSmax - X*exp(r*(q(i:M+1)-T));
               %boundary condition
         end
         V(:,i) = tempvec;
         D_Vec = interp1(s_,V(:,i-1),s,'spline')';
         numDivs = numDivs - 1;
         MatrixBoundary(N-1,i:M+1) = A(N)*V(N+1,i:M+1);
        end
    end
end
```

Listing 8.2: Matlab code fragment for interpolated option prices to account for dividends.

Figure 8.1 shows the surface of a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$ with two dividends. The first dividend is a fractional dividend at time $t = 0.25$ or 20 while the second is a 20% fractional dividend at time

$t = 0.75$. Option prices are interpolated in order to map to the spatial domain vector $\mathbf{s}$.

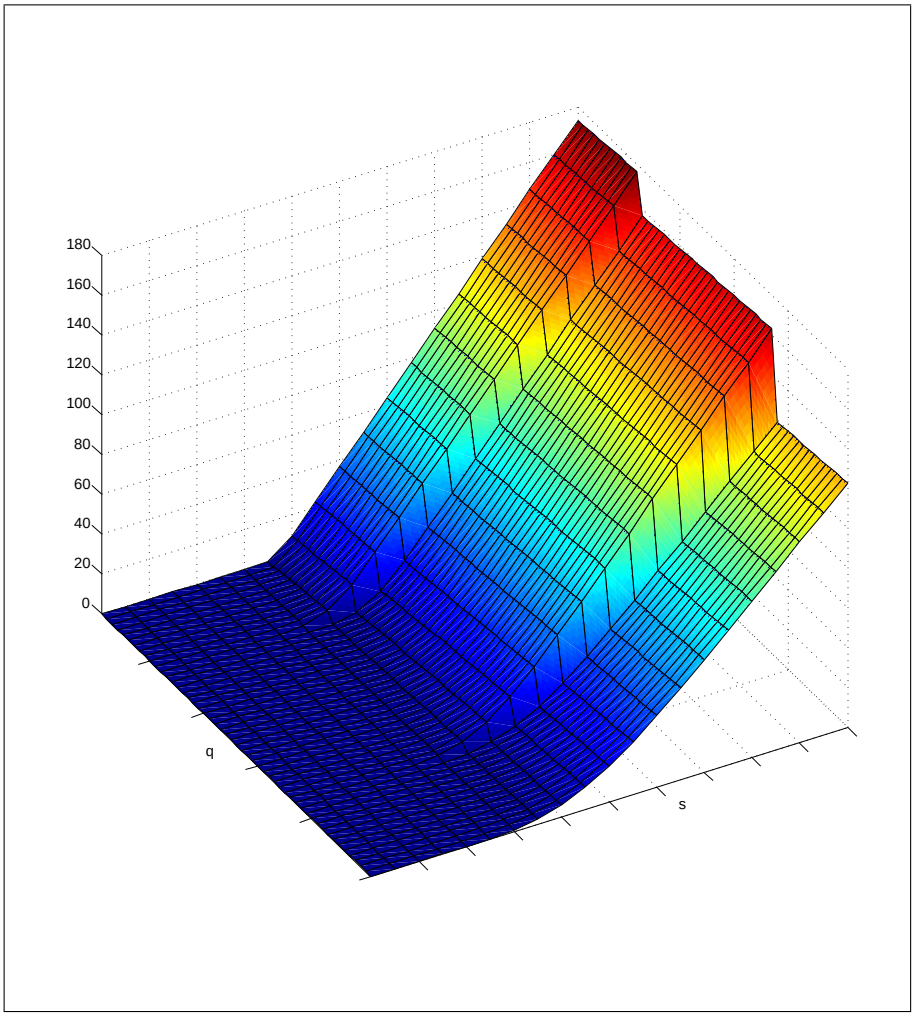

Figure 8.1: Du Fort and Frankel with interpolated option prices to account for dividends. Two dividends were used at time $\tau = 0.25$ (fractional dividend of 0.2) and time $\tau = 0.75$ (fixed dividend of 20) for an European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$.

8.5 Conclusion

The Black–Scholes–Merton stock price modeling environment does not provide for stock prices exhibiting dividend payments other than a continuous dividend yield. Efforts to analytically model options on stocks with discrete dividends prove futile, and despite a number of approximation techniques, generally numerical methods such as

the finite difference method are required to model the impact of dividend payments. In some cases analytical approximations are good, but since these models are without a sound theoretical foundation, care must be taken when using these. Analytical models are especially vulnerable when any form of path dependency in the option is present since these models generally model the timing of dividends poorly. Despite the obvious shortcomings of analytical dividend models, they are still widely used and are even incorporated in numeric techniques.

When options are not path dependant, fractional discrete dividends may be modeled analytically. For fixed discrete dividends and generally all discrete dividends when the contingent claim is path dependent, dividends should be modeled by assuming an arbitrage argument that states that the price of a contingent claim remains constant over the dividend period. It was shown that the basic pricing algorithm requires few adjustments to incorporate the arbitrage argument.

The Du Fort and Frankel scheme requires a slightly more complex treatment for dividends as this method requires two time steps to compute, thus computing prices one time step before an ex dividend date includes both ex– and cum–dividend prices. Adjustments must be made in order incorporate the correct underlying price vectors in the calculation. A further point often neglected in literature is the fact that the boundary conditions must also be adapted to cum-dividend prices when moving backwards in time from the ex dividend date to the cum dividend date.

Chapter 9

Discontinuous behavior

9.1 Introduction

A key characteristic common to a wide class of financial instruments is the presence of discontinuous behavior. An example is the payoff function of a European option given as

$$\Phi(S_T, T) = \max(S_T - X, 0),$$

where S_T is the underlying price at maturity and X is the strike price. The function is not differentiable with respect to S at the strike price at maturity. This discontinuous behavior not only puts a severe restriction on the attainable accuracy of a scheme [see for instance Linde et al., 2006], but also impacts on the estimation of other derivatives, notably

$$\frac{\partial^2 \hat{f}_j^i}{\partial q^2},$$

which is of special relevance to the Du Fort and Frankel scheme as it is a factor determining the severity of inconsistencies associated with the scheme. When $\frac{\partial^2 \hat{f}_j^i}{\partial q^2}$ becomes very large, as is the case around discontinuous regions, the scheme's consistency with the Black and Scholes partial differential equation is impacted leading to potentially invalid results.

The most common procedure to address areas of steep gradient is to refine the mesh around such points [Linde et al., 2006]. Sabau and Raad [1998] found that both high order compact schemes and low order classical schemes exhibit similar rates of convergence with uniform grid spacing and that refined grids are required in order to ob-

serve better convergence associated with the high order compact schemes in the regions where severe gradients are anticipated. Fornberg [1988] derives an algorithm which potentially utilizes the entire spatial vector in order to approximate derivatives of any order to a high degree of accuracy for any arbitrary grid points. The principal criticism of this technique is that the derivation of derivative in the more-dimensional case is uncertain. Linde et al. [2006] makes use of mesh overlapping in order to estimate high order derivatives around discontinuous areas, while Persson and Von Sydow [2007] adopt a two step mesh refinement algorithm whereby a quick solution is firstly found with a coarse grid. The solution then provide insight into areas where severe gradients occur, which is then solved with a refined grid around these areas.

We investigate the effectiveness of mesh refinements with vanilla European options, or in some cases barrier options. The Du Fort and Frankel finite difference scheme's results are compared to analytical results form Haug [1998].

9.2 Grid adjustment by analytic variable transformation

By transforming the Black and Scholes equation into the heat equation, one automatically assumes non constant spatial stepping in relation to the original spatial domain. Since the transform into the heat equation involves the transform $S = e^{\Xi\zeta}$, we are confronted with a transformed domain which has exponential spatial steps in the original spatial variable. The implication is that when contingent claims on the spatial domain are numerically estimated, relative more spatial steps are afforded to the lower part of the original domain than for the upper part. Such spacing matches the log-normal distribution of the underlying share price and it is well known that the stability properties of for instance the fully explicit method benefits markedly from log transforms [Tavella and Randall, 2000; Brennan and Schwartz, 1978].

A number of authors argue that a uniformly spaced (or log transformed) grids may not be the ideal configuration to achieve maximum accuracy. Since option prices often have steep gradients in their payoff functions (i.e. their initial boundary values), it appears logical to experiment with a spatial configuration that is more concentrated around the points were such gradients occur, for instance around the strike price of a European option.

Oosterlee et al. [2004], Clarke and Parrot [1999] and Leentvaar and Oosterlee [2006] employs a spatial transformation that concentrates grid points around the strike price of the option. Tavella and Randall [2000] warns that such transforms may impact on the convergence properties and other characteristics of the scheme. We thus exercise caution with the implementation of this transform on the Du Fort and Frankel scheme as it may impact on the seemingly vulnerable consistency properties of the scheme.
We base our approach on that of Oosterlee et al. [2004].
Instead of solving the function $\hat{f}(s,q) : \mathbb{R}^2 \to \mathbb{R}$, we solve a function on a transformed spatial grid, $\tilde{f}(\tilde{s},q) : \mathbb{R}^2 \to \mathbb{R}$, assuming $\hat{f}(s,q) = \tilde{f}(\tilde{s},q)$. The spatial domains stand in the functional relationship

$$\tilde{s} = \psi(s), \quad \text{and} \quad s = \varphi(\tilde{s}) = \psi^{-1}(\tilde{s}),$$

where the function $\psi(x)$ is arbitrarily given by

$$\psi(x) = \sinh^{-1}(\mu(x-\Lambda)) + \sinh^{-1}(\mu\Lambda), \tag{9.1}$$

but could take on many other forms. The coefficient μ is known as a stretching coefficient, the higher its value the more points around Λ there will be relatively to other regions. Λ is typically the strike price but could be set to any point where a concentration of grid points are required.
We construct a new grid with $\tilde{s} \in [\tilde{s}_\chi, \tilde{s}_\psi]$, uniformly spaced,

$$\tilde{h} = \tilde{s}_{j+1} - \tilde{s}_j = \frac{\tilde{s}_\psi - \tilde{s}_\chi}{N}; \qquad j = 2,3,\ldots,N,N+1.$$

By using the chain rule the Black and Scholes partial differential equation (equation 2.3) is transformed into

$$-\frac{\partial \hat{f}}{\partial q} + \frac{\alpha(\varphi(\tilde{s}))}{\varphi'(\tilde{s})}\frac{\partial}{\partial \tilde{s}}\left(\frac{1}{\varphi'(\tilde{s})}\frac{\partial \tilde{f}}{\partial \tilde{s}}\right) + \frac{\beta(\varphi(\tilde{s}))}{\varphi'(\tilde{s})}\frac{\partial \tilde{f}}{\partial \tilde{s}} + \gamma\varphi(\tilde{s})\hat{f} = 0. \tag{9.2}$$

Tavella and Randall [2000] discretises equation (9.2) directly. After algebraic manipulation the discrete Du Fort and Frankel scheme is written as

$$\tilde{f}_j^{i+1} = \xi(\tilde{A}\tilde{f}_{j+1}^i + \tilde{B}\tilde{f}_j^i + \tilde{C}\tilde{f}_{j-1}^i + \tilde{D}\tilde{f}_j^{i-1}),$$

where

$$\begin{aligned}
\tilde{A} &= 2\alpha(\varphi(\tilde{s}))k\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h}) + \beta(\varphi(\tilde{s}))kh\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h}),\\
\tilde{B} &= 2\gamma(\varphi(\tilde{s}))\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s})k\tilde{h}^2,\\
\tilde{C} &= 2\alpha(\varphi(\tilde{s}))k\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h}) - \beta(\varphi(\tilde{s}))kh\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h}),\\
\tilde{D} &= \varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s})\tilde{h}^2 - \alpha(\varphi(\tilde{s}))k(\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h}) + \varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})),\\
\xi &= (\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})\varphi'(\tilde{s})\tilde{h}^2 + \alpha(\varphi(\tilde{s}))k(\varphi'(\tilde{s}-\tfrac{1}{2}\tilde{h}) + \varphi'(\tilde{s}+\tfrac{1}{2}\tilde{h})))^{-1},\\
\tilde{h} &= N^{-1}(\psi(s_\psi) - \psi(s_\chi)).
\end{aligned}$$

Perhaps a more elegant procedure involves the transformation the functions $\alpha(s)$, $\beta(s)$ and $\gamma(s)$ [see for instance Oosterlee et al., 2004]. From equation (9.2)

$$\begin{aligned}
-\frac{\partial \hat{f}}{\partial q} + \frac{\alpha(\varphi(\tilde{s}))}{(\varphi'(\tilde{s}))^2}\frac{\partial^2 \tilde{f}}{\partial \tilde{s}^2}\cdots&\\
-\frac{\alpha(\varphi(\tilde{s}))\varphi''(\tilde{s})}{(\varphi(\tilde{s}))^3}\frac{\partial \tilde{f}}{\partial \tilde{s}} + \frac{\beta(\varphi(\tilde{s}))}{\varphi'(\tilde{s})}\frac{\partial \tilde{f}}{\partial \tilde{s}} + \gamma(\varphi(\tilde{s}))\tilde{f} &= 0\\
\therefore -\frac{\partial \tilde{f}}{\partial q} + \hat{\alpha}(\tilde{s})\frac{\partial^2 \tilde{f}}{\partial \tilde{s}^2} + \hat{\beta}(\tilde{s})\frac{\partial \tilde{f}}{\partial \tilde{s}} + \hat{\gamma}(\tilde{s})\tilde{f} &= 0, \qquad (9.3)
\end{aligned}$$

where

$$\begin{aligned}
\hat{\alpha}(\tilde{s}) &= \tfrac{\alpha(\varphi(\tilde{s}))}{(\varphi'(\tilde{s}))^2},\\
\hat{\beta}(\tilde{s}) &= \tfrac{\beta(\varphi(\tilde{s}))}{\varphi'(\tilde{s})} - \alpha(\varphi(\tilde{s}))\tfrac{\varphi''(\tilde{s})}{(\varphi'(\tilde{s}))^3},\\
\hat{\gamma}(\tilde{s}) &= \gamma(\varphi(\tilde{s})).
\end{aligned}$$

The original function is replaced by the transformed function. This procedure is elegant in the sense that employing different stretch functions becomes relatively simple in a computer coding sense. One simply replaces the function $\psi(x)$ with a different function. Listing segment (9.1) shows a Matlab implementation of the analytical grid refinement.

```
psi = @(x) (asinh(mu*(x-X_))+asinh(mu*X_));   %To transform s into y
phi = @(x) (1/mu * (sinh(x-asinh(mu*X_)))+X_);   %To transform y into s
phi_ = @(x) 1/mu * (cosh(x-asinh(mu*X_)));      %First derivative of phi
phi__ = @(x) 1/mu * (sinh(x-asinh(mu*X_)));     %Second derivative of phi

h = (psi(smax)-psi(smin))/N;
y = psi(smin):h:psi(smax);      %Transformed spatial variable
```

```
V = zeros(N+1,M+1);          %Solution matrix   initialization

%Assume U is known...
V(:,1:2) =U(:,1:2);
V(1,:) =0;
V(N+1,:) = (phi(y(N+1))-X*exp(r*(q-T)));

%FD Variables
a = @(x) 0.5.*o.*o.*(x.*x);
b = @(x) r.*x;
c = @(x) -r;

alpha = @(y) a(phi(y)).*( phi_(y).^2);
beta = @(y) b(phi(y)).* phi_(y) - a(phi(y)).* phi__(y).*( phi_(y).^3);
gamma = @(y) c(phi(y));
```

Listing 9.1: Matlab code segment for and analytic grid refinement.

9.2.1 Performance of grid adjustment for European options

The Du Fort and Frankel scheme performs poorly with the spatial transforms from equation (9.3). Figure (9.2) depicts the deviation of the Du Fort and Frankel finite difference solution from the analytic solution for a European option.

Although further research is required on the exact reasons for the poor performance, experimentation leads to us to believe that the equidistant spacing of the function $\psi(s)$ transformed back to the original spatial vector via the function $\varphi(\tilde{s})$ leads to relative fine spacing around the strike price. While this was the objective of the analytical grid stretching, schemes where the relationship between the spatial and temporal spacing impacts on the validity of the solution, such as the fully explicit scheme and the Du Fort and Frankel scheme, are potentially negatively impacted. Sottoriva and Rexhepi [2007] reports an inability to successfully implement a similar spatial transform with the explicit scheme. Although the nature of their difficulty is not revealed, our own experimentation with the explicit scheme indicates that implementation of the proposed

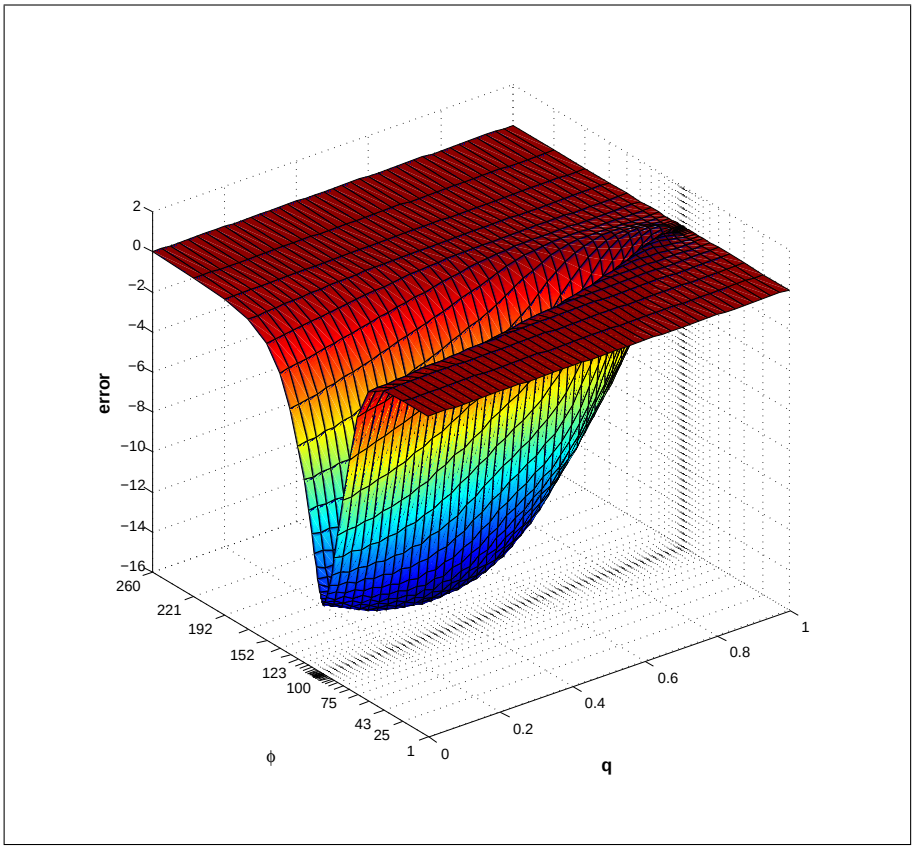

Figure 9.1: Error of the Du Fort and Frankel scheme with a stretched spatial variable ($\mu = 1$) for an European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$. The error compares the finite difference solution with an analytic solution.

spatial transform leads to severely impacted stability. This is probably due the the fact that the spatial spacing close to the strike price is roughly 25 times less than with the original spacing resulting in a far higher probability of instability.

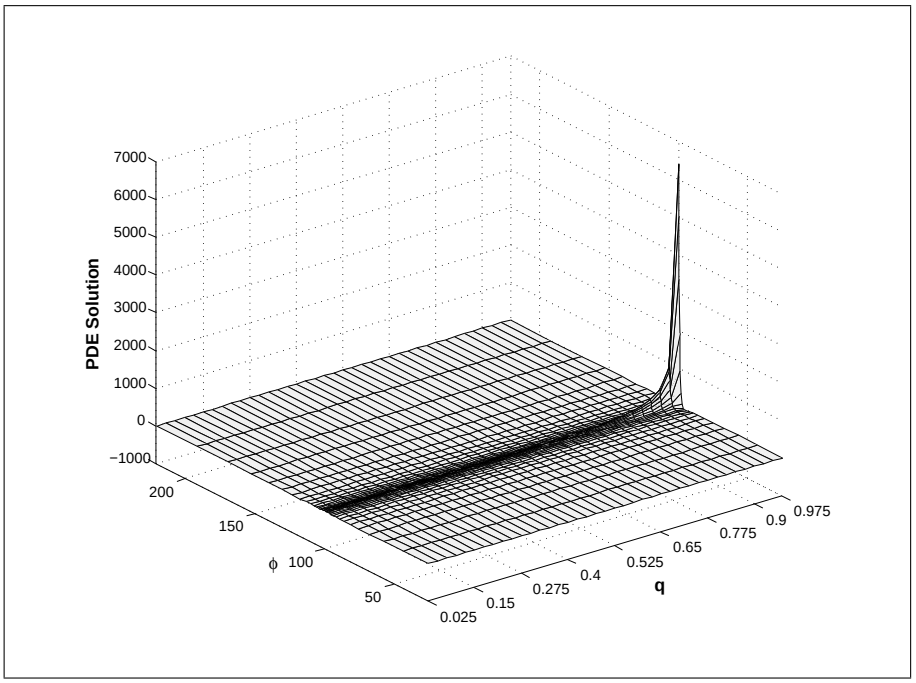

Figure 9.2: The solution of equation (9.3) using Du Fort and Frankel discretisation and a stretched spatial variable ($\mu = 1$) for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.15$ and $\sigma = 0.3$.

The Du Fort and Frankel scheme does not suffer from instability. Instead restrictions on the ratio of the spacing on the spatial axes in relation to the spacing on the temporal axes are required in order to be consistent with the partial differential equation. The finite difference scheme enforces the equality of equation (9.3). By substituting the derivatives of equation (9.3) with the relevant discrete approximations (see section (5.4.2)), and substituting the coefficients for their appropriate values, we found that the equality of equation (9.3) indeed holds. However, when we repeat the experiment with the analytic solution $f(\varphi(\tilde{s}), q)$, the equality does not hold, indicating the the finite difference scheme is not solving the function $f(\varphi(\tilde{s}), q)$. Figure (9.2) depicts the inconsistency by measuring by the amount that the partial differential equation deviates from zero.

9.2.2 Performance of grid adjustment for barrier options

Gatheral et al. [1999] reports improvement in accuracy by implementing a similar transform in order to solve barrier options using the Crank and Nicolson scheme. They use a function similar to equation (9.1) in order to achieve concentration of grid points around the level of the barrier. We use equation (9.1) as it is, setting $\Lambda = H$, the barrier level.

We test an up-and-out call option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$, $\sigma = 0.3$ and $H = 200$, and a down-and-out call option with similar input parameters and $H = 50$.

The up-and-out option performed poorly compared to the analytical solution published by Haug [1998]. The inconsistent term in the partial differential equation,

$$\frac{k^2}{h^2}\left(-\hat{\alpha}(\tilde{s})\frac{\partial^2 \tilde{f}}{\partial \tilde{s}^2}\right),$$

scales with values for $\hat{\alpha}(\tilde{s})$ which in turn scales with the underlying share price (see section (5.4.5)). Since h is very small around the barrier, we find a conspiracy of factors contributing to the invalidation of the result.

Figure (9.3) depicts the values of a numeric approximation of $|\frac{\partial^2 F}{\partial q^2}|$[1], k^2/h^2, where $h := \varphi(\tilde{s}(i+1)) - \varphi(\tilde{s}(i)), \quad i = 1, 2, \ldots, N-1, N$, and $\alpha(\tilde{s})$ for different values of the underlying instrument (top) and the deviation of the finite difference solution from the analytical solution (bottom). It is apparent from the graphic that the three elements contributing to inconsistency are rising in concert for underlying prices approaching the barrier level. The result is a solution that must be regarded as invalid.

In the case of the down and out barrier option, the results obtained were reasonable, although no real improvement was noted when comparing the solution of the adjusted spatial dimension with a solution with an unaltered spatial dimension.

Generally barrier options remain difficult to price with the Du Fort and Frankel scheme, and the utilization of analytic grid stretching proves unsuccessful.

[1] The value is approximated by

$$\frac{\partial^2 F}{\partial q^2} \approx \frac{1}{k^2}\left(F(\varphi(\tilde{s}), T-2k) - 2F(\varphi(\tilde{s}), T-k) + F(\varphi(\tilde{s}), T)\right),$$

where F is the analytic solution of the barrier option.

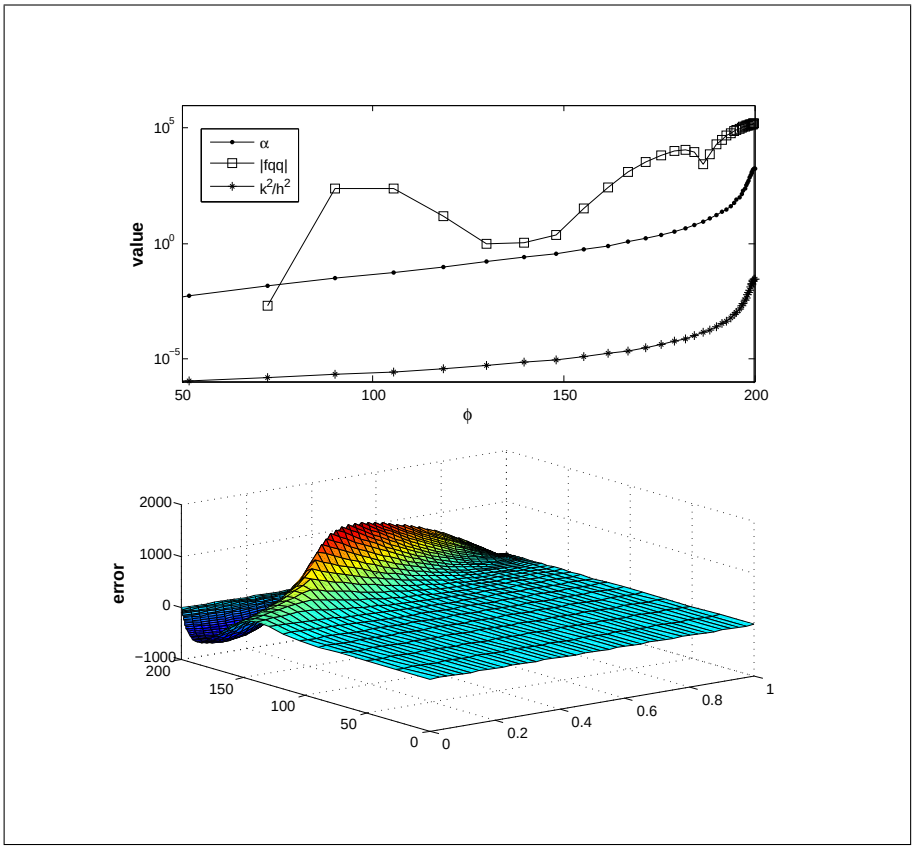

Figure 9.3: Du Fort and Frankel inconsistency with up-and-out barrier option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$, $\sigma = 0.3$ and $H = 200$. Values of the numeric approximation of $|\frac{\partial^2 F}{\partial q^2}|$, $\alpha(\tilde{s})$ and k^2/h^2 are depicted (top) with the deviation from the analytical solution (bottom).

9.3 Temporal grid adjustment

We experiment with an algorithm that refines the grid in the temporal direction. The algorithm results in a refined grid close to maturity that becomes coarser. Since the Du Fort and Frankel scheme requires two time steps in order to calculate the second order spatial derivative and temporal derivative, the algorithm is required to have spatial stepping such that for any time there must be two equidistant time steps that can be used in order to calculate these derivatives. This is achieved by defining a time step size $\mathbf{k} = (k(1), k(2), \ldots, k(\tilde{M})), \quad \tilde{M} \leq M,$ and $\tilde{M}$ is even. Every second time step in the vector $\mathbf{k}$ is double that of two time steps preceding it, i.e:

$$k(m) = 2k(m-2), \qquad m = 3, 4, \ldots, \tilde{M}.$$

Figure (9.4) schematically depicts the points utilized by the Du Fort and Frankel scheme for each time step.

The smallest step size, $\tilde{k} = k(1) = k(2)$ is given by

$$\tilde{k} = \frac{T}{2(2^{\frac{1}{2}\tilde{M}} - 1) + 2^{\frac{1}{2}\tilde{M}-1}(M - \tilde{M})}.$$

For $i = 3, 4, \ldots, \tilde{M}$, the step size $k(i)$ is given by

$$k(i) = \begin{cases} k(i-1) & \text{if } i \text{ is even, and} \\ 2k(i-1) & \text{if } i \text{ is odd.} \end{cases} \qquad i = 3, 4, \ldots, \tilde{M}.$$

The remaining step sizes $k(i), \quad i = \tilde{M}+1, \tilde{M}+2, \ldots, M$ remain constant, i.e. $k(i) = k(\tilde{M})$. The listing (9.2) shows a Matlab algorithm that computes the price of a European option with temporal adjustment.

```
clear;
clc;

S0 = 100;                    %Initial spot price
X = 100;                     %Strike price
T = 1.0;                     %Time to maturity
r = 0.1;                     %Risk free interest rate
o = 0.3;                     % Volatility
```

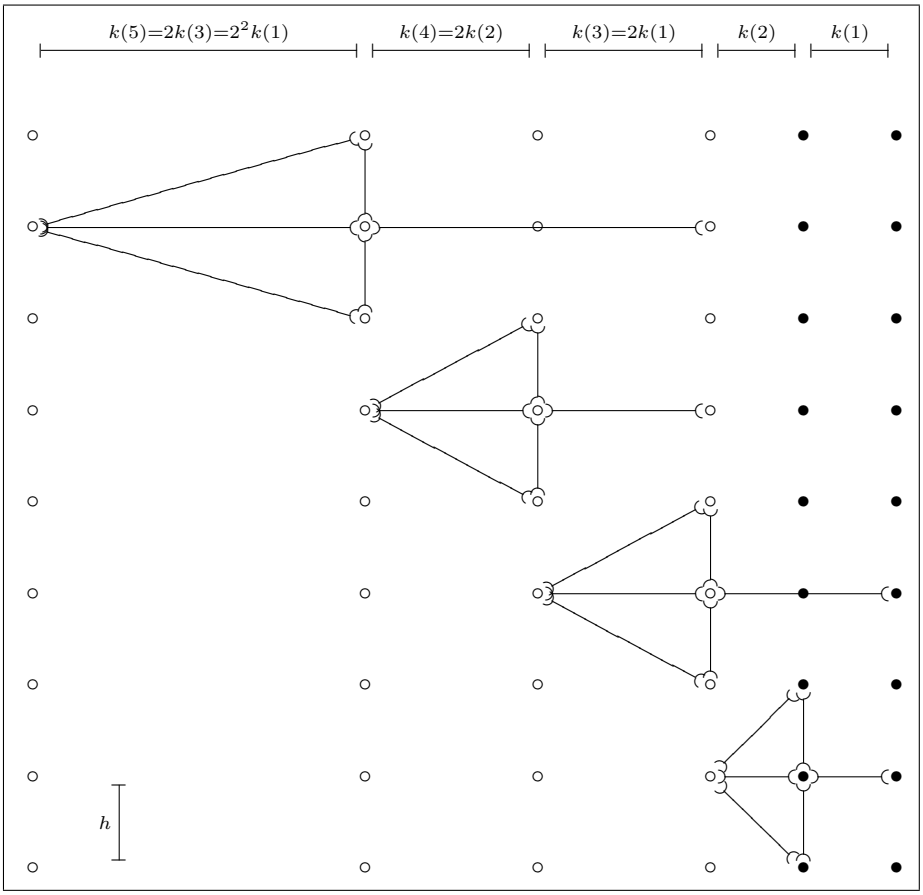

Figure 9.4: Du Fort and Frankel molecules for a refined grid in the temporal direction. Every second time step is twice that of two time steps before that. Boundary points are depicted by solid bullets

```
N = 20;        %Spatial steps
M = 20;        %Temporal steps
M_ =12;        %Refined Temporal steps

smin = 0;
smax = S0*exp((r-0.5*o*o)*T+4*o*sqrt(T));
NN = round(((X-smin)/(smax-smin))*N);
h = (X-smin)/NN;      %Spatial step size
s = smin+[0:N]*h;     %Spot prices
smax = s(N+1);

 %Adjusted Temporal step sizeq = zeros(M+1,1);
k_ = T/(2*(2^(0.5* M_)-1)+(M-M_)*2^(0.5*M_-1));
q = zeros(M+1,1);
q(M+1) = 0;
k = k_;
Odd = 1;
for i = M:-1:(M-M_+1)
    q(i,1) = q(i+1,1)+k;
    if Odd == -1
        k = k *2;
    end
    Odd = -Odd;
end
k = k/2;
for i = (M-M_):-1:1
    q(i,1) = q(i+1,1)+k;
end

k = flipdim(q(1:M)-q(2:M+1),1);

V = zeros(N+1,M+1);          %Solution matrix initialization
```

```
%European Option Dirichlet Boundary conditions
U = flipdim(bsmatrix('C', s,X,q,r,r,o),1)';
q = flipdim(q,1);
V(:,1:2) = U(:,1:2);
V(1,:) = 0;
V(N+1,:) = smax - X* exp(-r*(q));

alpha = @(x) 0.5*o^2*x.^2;
beta = @(x) r*x;
gamma = -r;

Odd = 1;

for i = 3:M+1
    A = (2*alpha(s)*k(i-1) + beta(s)*k(i-1)*h);
    B = (2*gamma*k(i-1)*h^2);
    C = (2*alpha(s)*k(i-1) - beta(s)*k(i-1)*h);
    D = (h^2 - 2*k(i-1)*alpha(s));

    if Odd == 1 || i > M_
        l = 2;
    else
        l = 3;
    end
    V(2:N,i) = ...
    (1./(h^2+2*alpha(s(2:N))*k(i-1)))'.*(A(2:N)'.*V(3:N+1,i-1)+...
        B*V(2:N,i-1)+C(2:N)'.*V(1:N-1,i-1)+D(2:N)'.*V(2:N,i-l));
    Odd = -Odd;
end
```

Listing 9.2: Matlab code for temporal refinement near the maturity date.

The algorithm listed in (9.2) does not impact materially on the accuracy of the scheme.

Figure (9.5) depicts three Du Fort and Frankel solutions for a European option with $X = 100$, $T = 1$, $r = 0.1$, $\sigma = 0.3$, $N = 20$, and $M = 20$. The depiction at the top (a) shows an unaltered grid, while the middle (b) and bottom graphics (c) depicts respectively 4 and 12 refined steps ($\tilde{M} = 4$, $\tilde{M} = 12$). No marked improvement is noted.

9.4 Adaptive mesh methods

Adaptive grids aim to control the error in the solution [Persson, 2006]. Errors are estimated and grid points along areas with high errors are increased such that the error is reduced to acceptable levels. By refining the grid one can expect the accuracy of a finite difference solution to improve, but at the cost of additional time to compute. The premise of adaptive mesh techniques is to only refine the mesh at the areas where accuracy is most severely impacted, typically where steep gradients occur. Numerous such techniques are described in literature. The estimation of the local error is subjective. Persson [2006] cites a local error of the form

$$e = \frac{||\frac{\partial \bar{F}}{\partial q} - \frac{F^n - F^{n+1}}{k}||_2}{||F^n||_2},$$

where $\frac{\partial \bar{F}}{\partial q}$ is a fourth order estimate of the time derivative and F^x is the estimate solution.

Fornberg [1988] describes an algorithm whereby space discretisations may be approximated for any arbitrarily spaced algorithm. Linde et al. [2006] note that it is unclear as to how such algorithm may be extended to the multidimensional case. Another disadvantage of the Fornberg [1988] algorithm is that space discretisation is required to take place in a single temporal vector. The Du Fort and Frankel finite difference scheme utilizes three time steps to calculate the diffusion process of the contingent claim. Utilizing more than one temporal vector with an arbitrarily spaced spatial vector is not clear from the algorithm.

Linde et al. [2006] presents a highly accurate scheme based on the overlay of multiple grids. They make use of two grids which are overlaid. A coarse grid is used to calculate the entire domain of the solution, while a finer grid is overlaid on specific regions where more mesh points are required. The coarse grid values are then replaced by the fine grid's values where they coincide before regressing through time. The method ap-

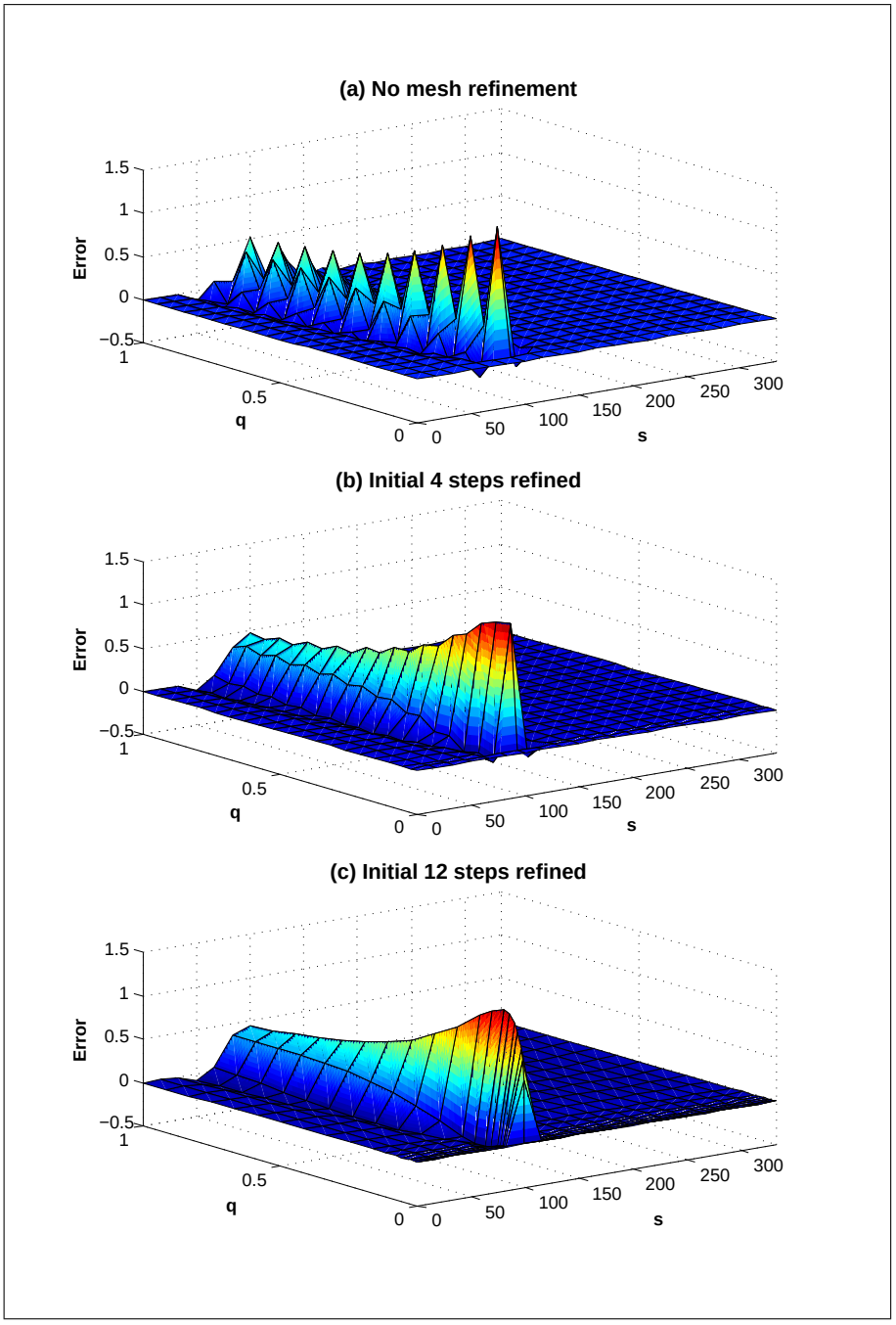

Figure 9.5: Du Fort and Frankel error for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$. The refined temporal steps are $\tilde{M} = 0$ at the top (a), $\tilde{M} = 4$ at the middle (b), and $\tilde{M} = 12$ at the bottom (c).

plies readily to the multi-dimensional case. A possible disadvantage of the Linde et al. [2006] algorithm is that it too is utilized on a single temporal vector at a time.

Figlewski and Gao [1999] classify errors associated with numerical schemes in two categories. The first is distribution error, which is the error of approximating the log-normal distribution on a discretised mesh. The second source of error is related to the non–linearity characteristics of the problem. The contingent claim is non–linear in the state variable in a way that cannot be captured by the discrete grid. The intention with the utilization of the adaptive mesh method is to minimize the non–linearity error of the scheme. Figlewski and Gao [1999] notes that although the adaptive mesh method is applied for a trinomial tree, such method can also be implemented for (explicit) finite difference methods since the methods are fundamentally similar. An attractive feature of the Figlewski and Gao [1999] method is that it is isomorphic at successive levels of refinement, i.e. that the same procedure can be implemented recursively to produce finer and finer grids around areas of problematic gradient.

The Figlewski and Gao [1999] method requires manual adjustment of the grid. Three different applications are discussed namely an American, a barrier and a general application to accurately compute the values of the trading greeks. In a subsequent paper Ahn et al. [1999] also discuss a procedure to build adaptive meshes for discrete barrier options. It was found that barrier options where the barrier level is discretely monitored potentially have values differing significantly from continuously monitored barriers.

Persson and Von Sydow [2007] makes use of a two step process. The first step calculates a solution by using a coarse grid. This solution provides insight as to place grid point in a more efficient way.

9.4.1 Limitations of the Du Fort and Frankel scheme

The Du Fort and Frankel scheme is severely limited in terms of grid adaptation as a result of its two time step structure. The refinement for the classical explicit scheme, similar to that described by Figlewski and Gao [1999] is depicted in figure (9.6).

Since the explicit scheme does not require upper and lower boundary conditions, the refinement is analogous to a trinomial tree. The grid points at time step $T - ik$ are assumed to be known. Typically these are boundary conditions. While this scheme works well for the classical explicit scheme (or trinomial trees), complications arise when two initialization steps in the temporal direction is required. This phenomenon is depicted

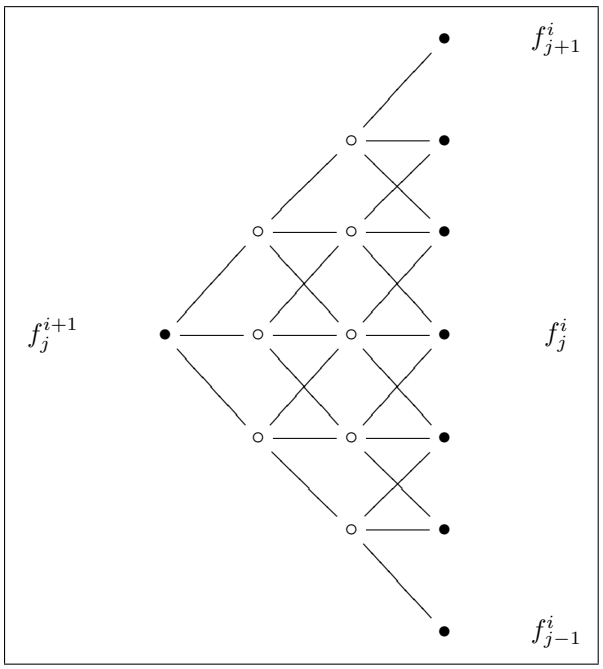

Figure 9.6: Grid refinement for an explicit scheme. Existing grid points are shown by solid bullets while the refined mesh is shown with hollow bullets.

in figure (9.7). Points between the time steps $T - ki$ and $T - k(i-1)$ are required. These are not readily derivable from the boundary conditions. We experimented with a technique that interpolates these intermediary points from other known points.

Interpolation of unknown intermediary boundary points

We experiment with an adaptive mesh technique that finds unknown points by interpolation. The technique makes use of an error estimate, $E \in 0, 1, 2, 3, 4 \ldots$. We make use of a coarse grid, and refine this around points where the expected error is highest by adopting a temporary spatial and temporal step size of $h(E) = 2^{-E}h$ and $k(E) = 2^{-E}k$ respectively. We arbitrarily choose an error estimate related to the second spatial derivative. From experimentation the error of the scheme appears to be related to the value of the second spatial derivative.

Figure (9.8) depicts the apparent relationship between the error of the Du Fort and Frankel scheme ($N = 50, M = 50$) for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$, and the gamma of the option (bottom). We arbitrarily

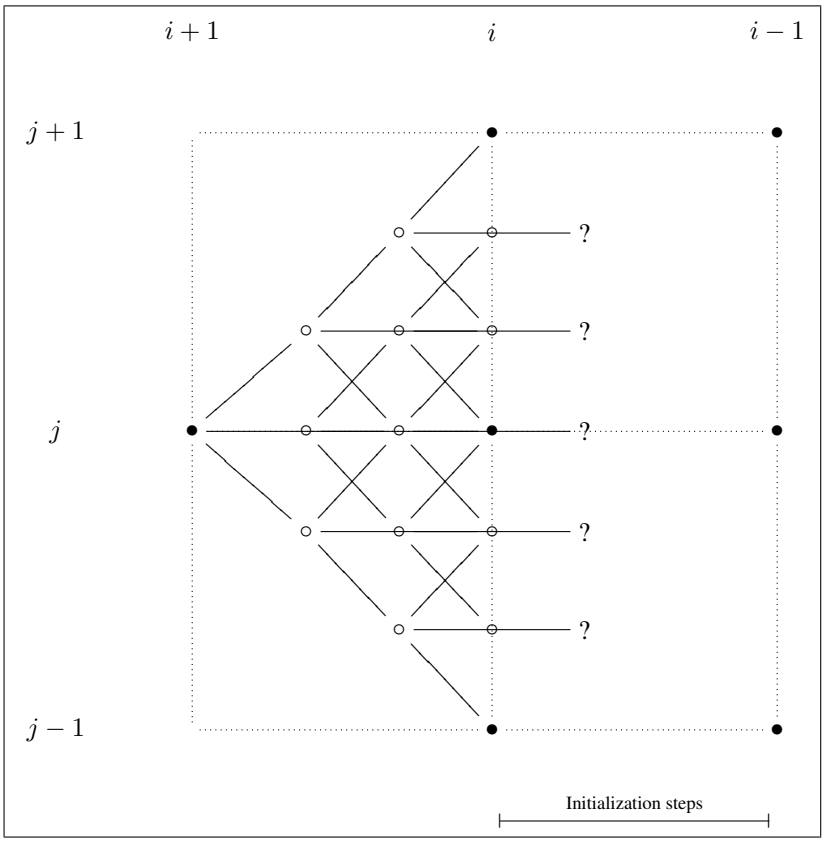

Figure 9.7: Grid refinement complications for the Du Fort and Frankel scheme. Existing grid points are shown by solid bullets while the refined mesh is shown with hollow bullets. Since two temporal steps are required in order to calculate each grid point, additional initialization steps are required between temporal steps $T - ki$ and $T - k(i-1)$.

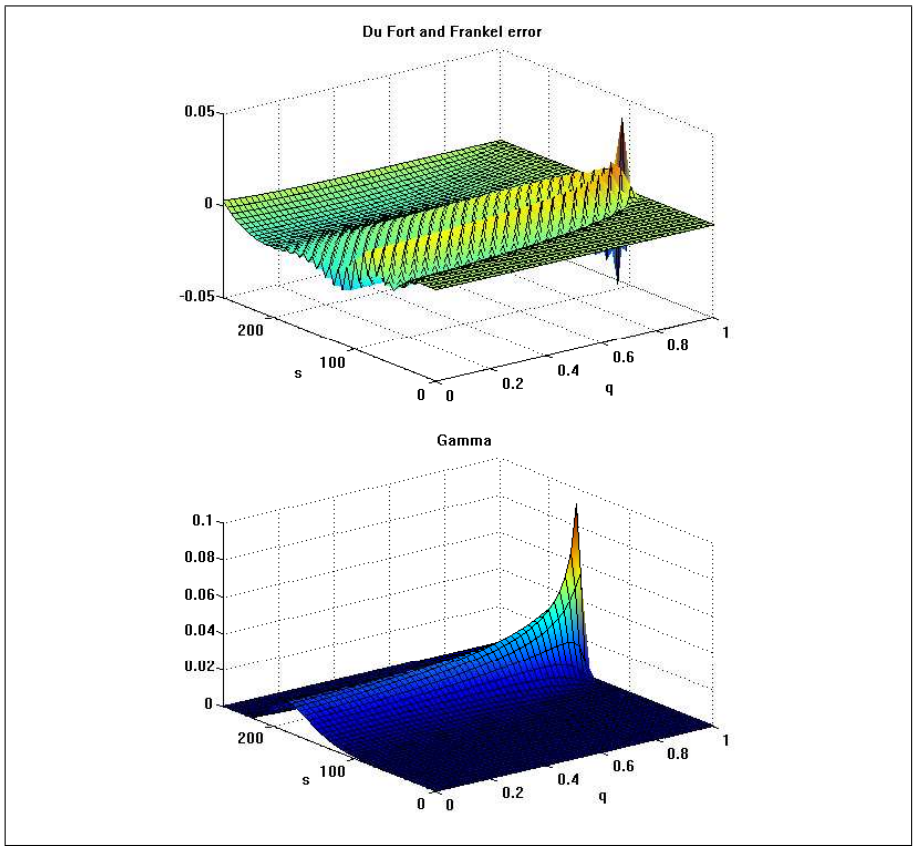

Figure 9.8: Du Fort and Frankel error for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$ (top) compared with the gamma of the same option (bottom).

chose the error estimate to be

$$E(j) = \text{Round}\left(\frac{\Sigma}{H}\max(\min(\tilde{E}(j), H+L) - L)\right),$$

where

$$\tilde{E}(j) = \frac{(\Gamma(j))^{\frac{1}{2}\sigma^2}}{\sqrt{T-t}},$$

with $\Gamma(j)$ being a numerical estimate of $\frac{\partial^2 f}{\partial s^2}$, i.e. $\Gamma = h^{-2}(f^i_{j+1} - 2f^i_j + f^i_{j-1})$. The parameter Σ is the maximum value for E, while the parameters H,L are subjectively chosen high and low values of Γ that correspond to $E = 0$ and $E = \Sigma$.

We assume all the temporal vectors $\mathbf{f}^\zeta, \quad \zeta = 1, 2, \ldots, i$ are known at the spatial

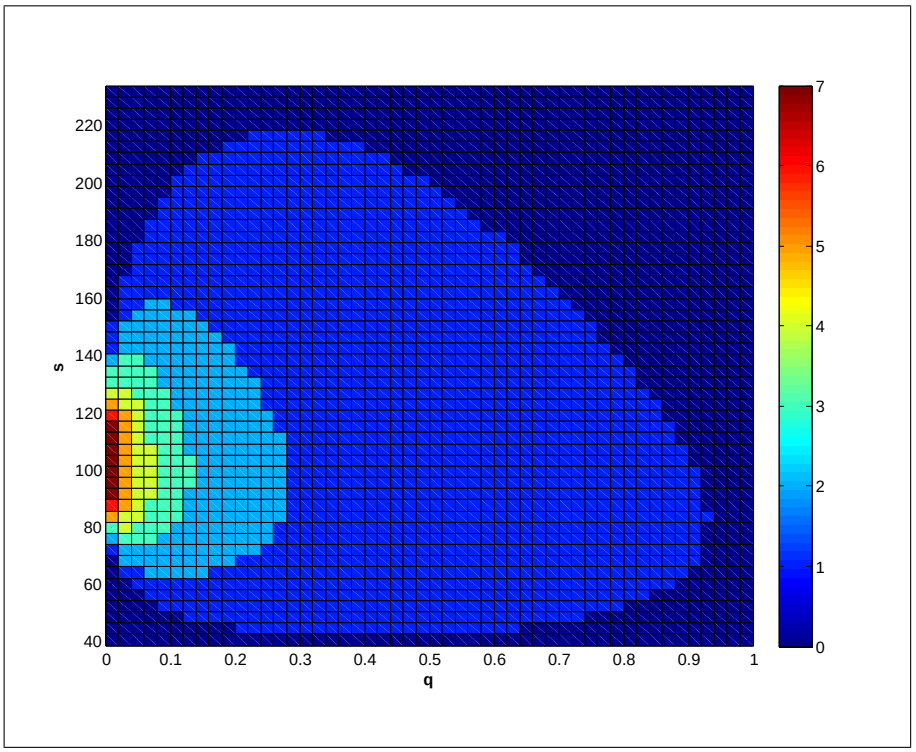

Figure 9.9: The error approximation for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$. The value of E ranges between $E \in [0, 7]$ with $H = 5$ and $L = 0.5$.

points $1, 2, \ldots, j-1, j, j+1, \ldots, N, N+1$, with f^i_1 and f^i_{N+1} trivially derivable from boundary conditions. The algorithm steps through the last known temporal vector, $\mathbf{f}^i$, estimating $E(j)$ for $j = 2, 3, \ldots, N$. For each $g = E(j), \ldots, 2, 1$, we estimate the

following grid points:

$$f(g) \in$$
$$\{f(ik, jh + h(E(j))), f(ik, jh), f(ik, jh - h(E(j))), \ldots$$
$$f(ik - k(E(j)), jh + h(E(j))), f(ik - k(E(j)), jh), \ldots$$
$$f(ik - k(E(j)), jh - h(E(j)))\}.$$

We make use of linear interpolation which is second order accurate. From $f(E(j))$, the points

$$\hat{f}(g) \in$$
$$\{f(ik + k(E(j)), jh + h(E(j))), f(ik + k(E(j)), jh), \ldots$$
$$f(ik + k(E(j)), jh - h(E(j)))\}$$

may be calculated. These points in turn are used to calculate the coarser values of $\hat{f}(g)$ corresponding to lower values of g. The Matlab code listed in listing 9.3 illustrates the algorithm.

```
clc ; clear ;
%Adaptive mesh with interpolated intermediate grid points
%Abie Bouwer 2007

%Input paramters
S0 = 100;
X = 100;
T = 1;
r = 0.1;
o = 0.3;

%Coarse Grid size
M = 19;
N = 19;
```

```
Segments = 7;
epsilon  = 3;

Smin = X*exp((r-0.5*o)*T-epsilon*o*sqrt(T));
Smax = X*exp((r-0.5*o)*T+epsilon*o*sqrt(T));

h = (Smax-Smin)/N;
s = Smin:h:Smax;
k = T/M;
t = T:-k:0;

High = 5;    %High Adjusted Gamma Value
Low = 0.5;   %Low Adjusted Gamma Value

ErrorMeasure = @(Gamma, Sigma, TimeToMaturity) ...
    Gamma.^(0.5*Sigma^2)./sqrt(TimeToMaturity');
nfunc = @(ErrorMeasure)
    round((max(min(ErrorMeasure,High+Low),Low)-Low)*Segments/High);

a = @(x) 0.5*o^2*x.^2;
b = @(x) r*x;
c = -r;

%Solution Matrix
U = bsmatrix('C',s,X,(T-t),r,r,o);
V = zeros(M+1,N+1);
V(1:2,:) = U(1:2,:);
V(:,1) = 0;
V(:,N+1) = Smax-X*exp(-r*(T-t));

%Temporary Localized Solution
A = zeros(3,3,8);
for i = 3:M+1
```

```
for j = 2:N
    %Populate A1
    A(1:2,1:3,1) = V(i-[2:-1:1],j+[-1:1]);

    E = nfunc(ErrorMeasure(...
    abs(h^-2*(V(i-1,j+1)-2*V(i-1,j)+V(i-1,j-1))),o,k*(i-1)));

    if E>0 %A(:,:,1) is already populated for E=0
        for l = 1:E
            %Alternative third order estimates inserted here ...
            A(2,2,l+1) = A(2,2,l);
            A(2,3,l+1) = 0.5*(A(2,3,l)+A(2,2,l));
            A(2,1,l+1) = 0.5*(A(2,1,l)+A(2,2,l));
            A(1,2,l+1) = 0.5*(A(2,2,l)+A(1,2,l));
            A(1,3,l+1) = 0.25*(A(2,3,l)+A(1,3,l)+A(2,2,l)+A(1,2,l));
            A(1,1,l+1) = 0.25*(A(2,1,l)+A(1,1,l)+A(2,2,l)+A(1,2,l));
            %... to here ...
        end
        for l = E:-1:1
            h_ = h*2^(-l);
            k_ = k*2^(-l);
            if l==E
    A(3,3,l+1) = ...
A(2,3,l)*((2*a(s(j)+h_)*k_+b(s(j)+h_)*k_*h_)/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(2,3,l+1)*((2*c*k_*h_^2)/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(2,2,l)*(((2*a(s(j)+h_)*k_-b(s(j)+h_)*k_*h_))/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(1,3,l+1)*((h_^2-2*a(s(j)+h_)*k_)/(h_^2+2*a(s(j)+h_)*k_));

    A(3,2,l+1) = ...
A(2,3,l+1)*((2*a(s(j))*k_+b(s(j))*k_*h_)/(h_^2+2*a(s(j))*k_)) + ...
A(2,2,l)*((2*c*k_*h_^2)/(h_^2+2*a(s(j))*k_)) + ...
A(2,1,l+1)*(((2*a(s(j))*k_-b(s(j))*k_*h_))/(h_^2+2*a(s(j))*k_)) + ...
A(1,2,l+1)*((h_^2-2*a(s(j))*k_)/(h_^2+2*a(s(j))*k_));
```

```
        A(3,1,l+1) = ...
A(2,2,l)*((2*a(s(j)-h_)*k_+b(s(j)-h_)*k_*h_)/(h_^2+2*a(s(j)-h_)*k_)) + ...
A(2,1,l+1)*((2*c*k_*h_^2)/(h_^2+2*a(s(j)-h_)*k_)) + ...
A(2,1,l)*(((2*a(s(j)-h_)*k_-b(s(j)-h_)*k_*h_))/(h_^2+2*a(s(j)-h_)*k_)) + ...
A(1,1,l+1)*((h_^2-2*a(s(j)-h_)*k_)/(h_^2+2*a(s(j)-h_)*k_));
                else
        A(3,3,l+1) = ...
A(2,3,l)*((2*a(s(j)+h_)*k_+b(s(j)+h_)*k_*h_)/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(2,3,l+1)*((2*c*k_*h_^2)/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(2,2,l)*(((2*a(s(j)+h_)*k_-b(s(j)+h_)*k_*h_))/(h_^2+2*a(s(j)+h_)*k_)) + ...
A(1,3,l+1)*((h_^2-2*a(s(j)+h_)*k_)/(h_^2+2*a(s(j)+h_)*k_));

        A(3,2,l+1) = ...
A(3,3,l+2)*((2*a(s(j))*(k_/2)+b(s(j))*(k_/2)*...
    (h_/2))/((h_/2)^2+2*a(s(j))*(k_/2))) + ...
A(3,2,l+2)*((2*c*(k_/2)*(h_/2)^2)/((h_/2)^2+...
    2*a(s(j))*(k_/2))) + ...
A(3,1,l+2)*(((2*a(s(j))*(k_/2)-b(s(j))*(k_/2)*...
    (h_/2)))/((h_/2)^2+2*a(s(j))*(k_/2))) + ...
A(2,2,l+2)*(((h_/2)^2-2*a(s(j))*(k_/2))/...
    ((h_/2)^2+2*a(s(j))*(k_/2)));

        A(3,1,l+1) = ...
A(2,2,l)*((2*a(s(j)-h_)*k_+b(s(j)-h_)*k_*h_)/...
    (h_^2+2*a(s(j)-h_)*k_)) + ...
A(2,1,l+1)*((2*c*k_*h_^2)/...
    (h_^2+2*a(s(j)-h_)*k_)) + ...
A(2,1,l)*(((2*a(s(j)-h_)*k_-b(s(j)-h_)*k_*h_))/...
    (h_^2+2*a(s(j)-h_)*k_)) + ...
A(1,1,l+1)*((h_^2-2*a(s(j)-h_)*k_)/...
    (h_^2+2*a(s(j)-h_)*k_));
                end
```

```
            end
            h_ = h/2;
            k_ = k/2;
             V(i,j) = ...
A(3,3,2)*((2*a(s(j))*k_+b(s(j))*k_*h_)/(h_^2+2*a(s(j))*k_)) + ...
A(3,2,2)*((2*c*k_*h_^2)/(h_^2+2*a(s(j))*k_)) + ...
A(3,1,2)*(((2*a(s(j))*k_-b(s(j))*k_*h_))/(h_^2+2*a(s(j))*k_)) + ...
A(2,2,2)*((h_^2-2*a(s(j))*k_)/(h_^2+2*a(s(j))*k_));
        else
            V(i,j) = ...
V(i-1,j+1)*((2*a(s(j))*k+b(s(j))*k*h)/(h^2+2*a(s(j))*k)) + ...
V(i-1,j)*((2*c*k*h^2)/(h^2+2*a(s(j))*k)) + ...
V(i-1,j-1)*(((2*a(s(j))*k-b(s(j))*k*h))/(h^2+2*a(s(j))*k)) + ...
V(i-2,j)*((h^2-2*a(s(j))*k)/(h^2+2*a(s(j))*k));
        end
    end
end
```

Listing 9.3: Matlab code for interpolation technique combined with a adaptive mesh.

The performance of the second order interpolation is disappointing. Figure (9.10) depicts the error made with the linear grid refinement technique.

We improve the algorithm by improving the interpolation accuracy to third order. Figure (9.11) depicts the fictitious points inserted on the grid. We interpolate values for the additional points (indicated with stars ($\star$)) by making use of third order accurate estimations for the partial derivatives $\frac{\partial^2 f_y^x}{\partial s^2}$, $\frac{\partial f_y^x}{\partial s}$ and $\frac{\partial f_y^x}{\partial q}$, where x and y are general coordinates on the grid.

The spatial derivatives are found from manipulating Taylor expansions, while the temporal derivative is derived from the Black–Scholes partial differential equation (equation 2.3). The following Tailor expansions are used in the determination of the spatial derivatives, where $\bar{h}$ and $\bar{k}$ are general spatial step sizes, and f_s, f_{ss} etc. are

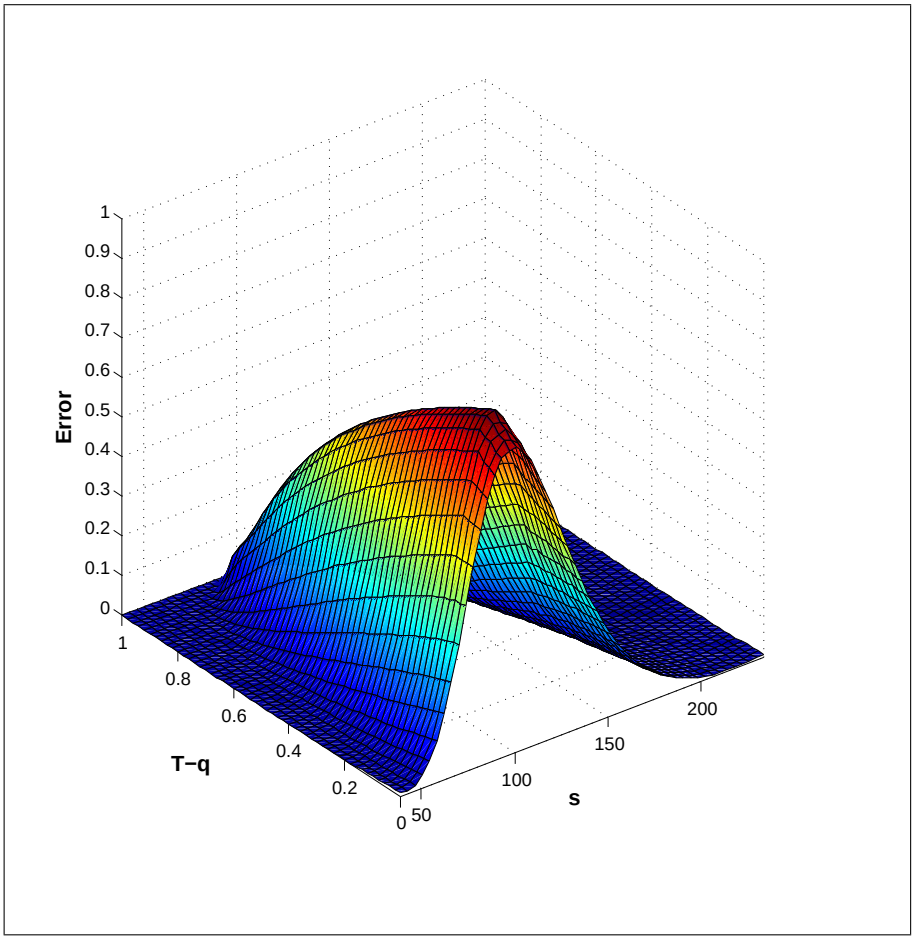

Figure 9.10: The difference between the Du Fort and Frankel scheme and the analytic solution for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$. A linear interpolated grid refinement technique was used.

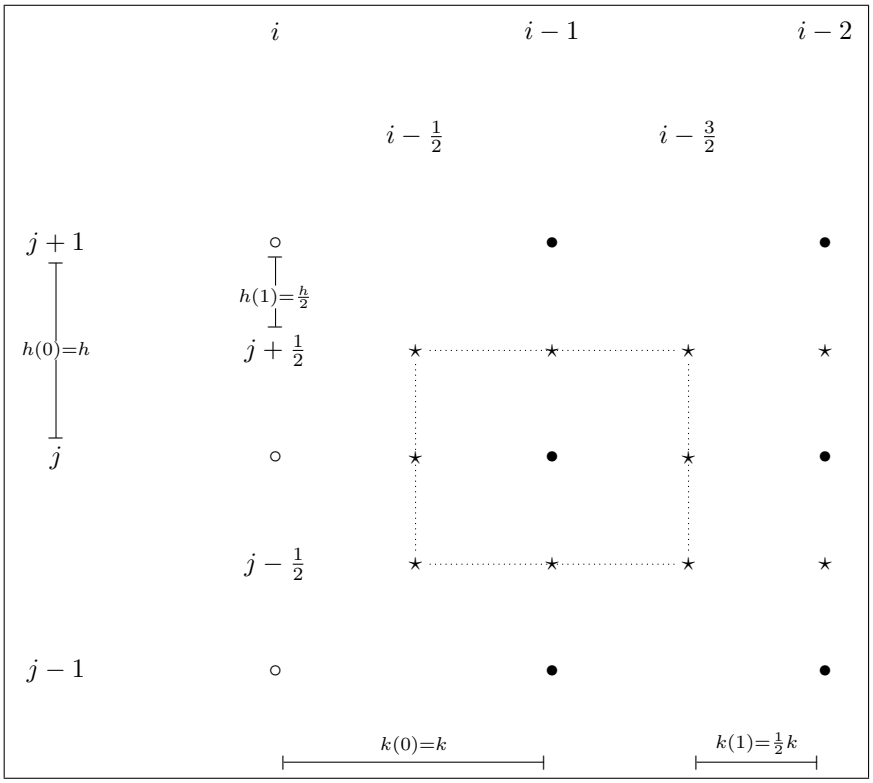

Figure 9.11: Schematic representation of a single grid refinement step. Fictitious points indicated with stars ($\star$) are interpolated from the original points indicated by bullets ($\bullet$) for known points and circles ($\circ$) for unknown points.

shorthand for the first, second, etc. spatial derivatives in the point $(\bar{h}y, \bar{k}x)$:

$$\begin{aligned}
\text{(a)}\quad f^x_{y+1} &= f^x_y + \bar{h}f_s + \frac{1}{2}\bar{h}^2 f_{ss} + \frac{1}{6}\bar{h}^3 f_{sss} + \frac{1}{24}\bar{h}^4 f_{ssss} + \dots \\
\text{(b)}\quad f^x_{y-1} &= f^x_y - \bar{h}f_s + \frac{1}{2}\bar{h}^2 f_{ss} - \frac{1}{6}\bar{h}^3 f_{sss} + \frac{1}{24}\bar{h}^4 f_{ssss} + \dots \\
\text{(c)}\quad f^x_{y+2} &= f^x_y + 2\bar{h}f_s + \frac{4}{2}\bar{h}^2 f_{ss} + \frac{8}{6}\bar{h}^3 f_{sss} + \frac{16}{24}\bar{h}^4 f_{ssss} + \dots \\
\text{(d)}\quad f^x_{y-2} &= f^x_y - 2\bar{h}f_s + \frac{4}{2}\bar{h}^2 f_{ss} - \frac{8}{6}\bar{h}^3 f_{sss} + \frac{16}{24}\bar{h}^4 f_{ssss} + \dots
\end{aligned}$$

Simple algebraic manipulation reveals values for f_s and f_{ss} for the point $(jh(0), (i-1)k(0))$:

$$\begin{aligned}
\Gamma^{i-1}_j \equiv f_{ss} &= \frac{1}{12\bar{h}^2}(-f^{i-1}_{j+2} + 16f^{i-1}_{j+1} - 30f^{i-1}_j + 16f^{i-1}_{j-1} - f^{i-1}_{j-2}) + \mathcal{O}(\bar{h}^3) \\
\Delta^{i-1}_j \equiv f_s &= \frac{1}{6\bar{h}}(-f^{i-1}_{j+2} + 6f^{i-1}_{j+1} - 3f^{i-1}_j - 2f^{i-1}_{j-1}) + \mathcal{O}(\bar{h}^3)
\end{aligned}$$

The remaining derivatives are found in similar way for the remaining points $((jh(0) \pm h(1), (i-1)k(0) \pm k(1))$.

The temporal derivative in point $(jh(0), (i-1)k(0))$ is found by substituting the estimates for $\frac{\partial f}{\partial s}$ and $\frac{\partial^2 f}{\partial s^2}$ into the Black–Scholes partial differential equation (equation 2.3):

$$\Theta^{i-1}_j \equiv f_t = \frac{1}{2}\sigma^2(s(j))^2\Gamma^{i-1}_j + r(s(j))\Delta^{i-1}_j - rf^{i-1}_j + \mathcal{O}(h^3).$$

The temporal derivatives for the remaining points are found in similar fashion.
With the temporal derivative known, we interpolate point $(jh(0), (i-\frac{2}{3})k(0))$ from known points $(jh(0), (i-1)k(0))$ and $(jh(0), (i-2)k(0))$ by making use of the following Taylor expansions:

$$\begin{aligned}
\text{(a)}\quad f^{i-2}_j &= f^{i-\frac{3}{2}}_j - k(1)f_t + \frac{1}{2}(k(1))^2 f_{tt} - \frac{1}{6}(k(1))^3 f_{ttt} + \mathcal{O}((k(1))^4) \\
\text{(b)}\quad f^{i-\frac{7}{4}}_j &= f^{i-\frac{3}{2}}_j - \frac{1}{2}k(1)f_t + \frac{1}{8}(k(1))^2 f_{tt} - \frac{1}{48}(k(1))^3 f_{ttt} + \mathcal{O}((k(1))^4) \\
\text{(c)}\quad f^{i-1}_j &= f^{i-\frac{3}{2}}_j + k(1)f_t + \frac{1}{2}(k(1))^2 f_{tt} + \frac{1}{6}(k(1))^3 f_{ttt} + \mathcal{O}((k(1))^4) \\
\text{(d)}\quad f^{i-\frac{5}{4}}_j &= f^{i-\frac{3}{2}}_j - \frac{1}{2}k(1)f_t + \frac{1}{8}(k(1))^2 f_{tt} - \frac{1}{48}(k(1))^3 f_{ttt} + \mathcal{O}((k(1))^4),
\end{aligned}$$

we obtain

$$\begin{aligned}
f^{i-2}_j - 4f^{i-\frac{7}{4}}_j - 4f^{i-1}_j + f^{i-\frac{5}{4}}_j &= -6f^{i-\frac{3}{2}}_j + \mathcal{O}((k(1))^3) \\
\therefore f^{i-\frac{3}{2}}_j &= -\frac{1}{6}(f^{i-2}_j + f^{i-1}_j) + \frac{2}{3}(f^{i-\frac{7}{4}}_j + f^{i-\frac{5}{4}}_j) + \mathcal{O}((k(1))^3). \qquad (9.4)
\end{aligned}$$

Additional Taylor expansion yields

$$\begin{aligned} f_j^{i-\frac{5}{4}} &= f_j^{i-2} + \frac{3}{2}k(1)u_t + \frac{9}{8}(k(1))^2 u_t t + \mathcal{O}((k(1))^3) \\ f_j^{i-\frac{7}{4}} &= f_j^{i-2} + \frac{1}{2}k(1)u_t + \frac{1}{8}(k(1))^2 u_t t + \mathcal{O}((k(1))^3) \\ \therefore f_j^{i-\frac{7}{4}} - 9f_j^{i-\frac{5}{4}} &= -8f_j^{i-2} - 3k(0)u_t + \mathcal{O}((k(1))^3), \end{aligned} \tag{9.5}$$

and similarly

$$\begin{aligned} f_j^{i-\frac{5}{4}} &= f_j^{i-1} - \frac{1}{2}k(1)u_t + \frac{1}{8}(k(1))^2 u_t t + \mathcal{O}((k(1))^3) \\ f_j^{i-\frac{7}{4}} &= f_j^{i-1} - \frac{3}{2}k(1)u_t + \frac{9}{8}(k(1))^2 u_t t + \mathcal{O}((k(1))^3) \\ \therefore f_j^{i-\frac{5}{4}} - 9f_j^{i-\frac{7}{4}} &= -8f_j^{i-1} + 3k(0)u_t + \mathcal{O}((k(1))^3). \end{aligned} \tag{9.6}$$

Adding equation (9.5) to equation (9.6) gives

$$f_j^{i-\frac{5}{4}} + f_j^{i-\frac{7}{4}} = f_j^{i-2} + f_j^{i-1} + \frac{3}{8}(\Theta_j^{i-2} - \Theta_j^{i-1}) + \mathcal{O}((k(1))^3). \tag{9.7}$$

Substituting equation (9.7) into equation (9.4) provides an estimate for the point $(jh(0), (i-\frac{3}{2})k(0))$:

$$f_j^{i-\frac{3}{2}} = \frac{1}{2}(f_j^{i-2} + f_j^{i-1}) + \frac{k(1)}{4}(\Theta_j^{i-2} - \Theta_j^{i-1}) + \mathcal{O}((k(1))^3). \tag{9.8}$$

The points $((j \pm \frac{1}{2})h(0), (i-2)k(0))$ are interpolated by adding multiples of equations from the following Taylor expansions:

$$\text{(a)} \quad f_{j+1}^{i-2} = f_{j+\frac{1}{2}}^{i-2} + h(1)\frac{\partial f_{j+\frac{1}{2}}^{i-2}}{\partial s} + \frac{1}{2}(h(1))^2\frac{\partial^2 f_{jj+\frac{1}{2}}^{i-2}}{\partial s^2} + \frac{1}{6}(h(1))^3\frac{\partial^3 f_{j+\frac{1}{2}}^{i-2}}{\partial s^3} + \mathcal{O}((h(1))^4)$$

$$\text{(b)} \quad f_{j}^{i-2} = f_{j+\frac{1}{2}}^{i-2} - h(1)\frac{\partial f_{j+\frac{1}{2}}^{i-2}}{\partial s} + \frac{1}{2}(h(1))^2\frac{\partial^2 f_{j+\frac{1}{2}}^{i-2}}{\partial s^2} - \frac{1}{6}(h(1))^3\frac{\partial^3 f_{j+\frac{1}{2}}^{i-2}}{\partial s^3} + \mathcal{O}((h(1))^4)$$

$$\text{(c)} \quad f_{j-2}^{i-2} = f_{j+\frac{1}{2}}^{i-2} - 3h(1)\frac{\partial f_{j+\frac{1}{2}}^{i-2}}{\partial s} + \frac{9}{2}(h(1))^2\frac{\partial^2 f_{j+\frac{1}{2}}^{i-2}}{\partial s^2} - \frac{27}{6}(h(1))^3\frac{\partial^3 f_{j+\frac{1}{2}}^{i-2}}{\partial s^3} + \mathcal{O}((h(1))^4)$$

The addition of $3 \times$ (a) $+ 6 \times$ (b) $-$ (c) yields

$$f_{j+\frac{1}{2}}^{i-2} = \frac{1}{8}(3f_{j+1}^{i-2} + 6f_j^{i-2} - f_{j-2}^{i-2}).$$

The function f in the point $(j - \frac{1}{2}h(0), (i-2)k(0))$ is found in similar fashion. These two approximations are then used in order to estimate the function f in the points $((j \pm \frac{1}{2})h(1), (i-2)k(0))$, by employing a similar technique than in equation (9.8).

The algorithm in listing (9.3) is easily adapted to incorporate the more accurate interpolations. By replacing the indicated lines in listing (9.3) by the code fragment listed in listing (9.4), we obtain third order accurate interpolations.

```
h_ = h*2^(-l+1);
k_ = k*2^(-l+1);
A(2,2,l+1) = A(2,2,l);
A(2,3,l+1) = 0.125*(3*A(2,3,l)+6*A(2,2,l)-A(2,1,l)); %O(h3)
A(2,1,l+1) = 0.125*(3*A(2,1,l)+6*A(2,2,l)-A(2,3,l)); %O(h3)
p1 = 0.125*(3*A(1,3,l)+6*A(1,2,l)-A(1,1,l)); %O(h3)
p2 = 0.125*(3*A(1,1,l)+6*A(1,2,l)-A(1,3,l)); %O(h3)
g1 = 1/(12*h_^2)*(11*A(2,3,l)+6*A(2,2,l)+...
     4*A(2,1,l+1)-A(2,1,l)-20*A(2,3,l+1));
g2 = 1/(12*h_^2)*(11*A(2,1,l)+6*A(2,2,l)+...
     4*A(2,3,l+1)-A(2,3,l)-20*A(2,1,l+1));
g3 = 1/(12*h_^2)*(11*A(1,3,l)+6*A(1,2,l)+...
     4*p2 -A(1,1,l)-20*p1);
g4 = 1/(12*h_^2)*(11*A(1,1,l)+6*A(1,2,l)+...
     4*p1-A(2,3,l)-20*p2);
g5 = 1/(12*h_^2)*(-A(2,1,l)+16*A(2,1,l+1)-...
     30*A(2,2,l)+16*A(2,3,l+1)-A(2,3,l));
g6 = 1/(12*h_^2)*(-A(1,1,l)+16*p2-...
     30*A(1,2,l)+16*p1-A(1,3,l));
d1 = 1/(6*h_)*(-2*A(2,3,l)+6*A(2,2,l)-...
     A(2,1,l+1)-3*A(2,3,l+1));
d2 = 1/(6*h_)*(-2*A(2,1,l)+6*A(2,2,l)-...
     A(2,3,l+1)-3*A(2,1,l+1));
d3 = 1/(6*h_)*(-2*A(1,3,l)+6*A(1,2,l)-p2-3*p1);
d4 = 1/(6*h_)*(-2*A(1,1,l)+6*A(1,2,l)-p1-3*p2);
d5 = 1/(6*h_)*(-A(2,3,l)+6*A(2,3,l+1)-...
     3*A(2,2,l)-2*A(2,1,l+1));
d6 = 1/(6*h_)*(-A(1,3,l)+6*p1-3*A(1,2,l)-2*p2);
t1 = a(s(j)+h_)*g1+b(s(j)+h_)*d1 + c*A(2,3,l+1);
```

```
t2 = a(s(j)−h_)∗g2+b(s(j)−h_)∗d2 + c∗A(2,1,1+1);
t3 = a(s(j)+h_)∗g3+b(s(j)+h_)∗d3 + c∗p1;
t4 = a(s(j)−h_)∗g4+b(s(j)−h_)∗d4 + c∗p2;
t5 = a(s(j))∗g5+b(s(j))∗d5 + c∗A(2,2,1+1);
t6 = a(s(j))∗g6+b(s(j))∗d6 + c∗A(1,2,1);
A(1,3,1+1) = 0.5∗(A(2,3,1+1)+p1)+0.25∗k_∗(t3−t1);
A(1,2,1+1) = 0.5∗(A(2,2,1)+A(1,2,1))+0.25∗k_∗(t6−t5);
A(1,1,1+1) = 0.5∗(A(2,1,1+1)+p2)+0.25∗k_∗(t4−t2);
```

Listing 9.4: Matlab code segment to improve interpolation to third order.

Results for numerical experimentation with the third order interpolated grid refinement algorithm are encouraging, but not entirely unproblematic. With a low number of grid points ($N = 19, M = 19$) the interpolated grid refinement algorithm produces vastly superior results to a similar unaltered grid. The result is depicted in figure (9.12). However, the interpolated grid seem to be unstable, and for a higher number of grid points, the number of time steps in relation to spatial steps need to increase. Further research in relation to the required ratios between time steps and spatial steps is required.

9.5 Summary of measures to improve numerical performance

Table 9.1 summarizes the measures we investigated to improve numerical performance at areas where steep gradients occur.

9.6 Conclusion

In this chapter we considered various techniques that alleviate the problems associated with discontinuous behavior in the price of a contingent claim. The Du Fort and Frankel scheme pose a special challenges in this regard for two reasons: The first is that care must be exercised with relation to the ratio of step sizes in the spatial and temporal directions. The Du Fort and Frankel scheme is known to be inconsistent with the Black and Scholes partial differential equation if the spatial step size becomes small in relation to the temporal step size. The second reason is that the Du Fort and Frankel

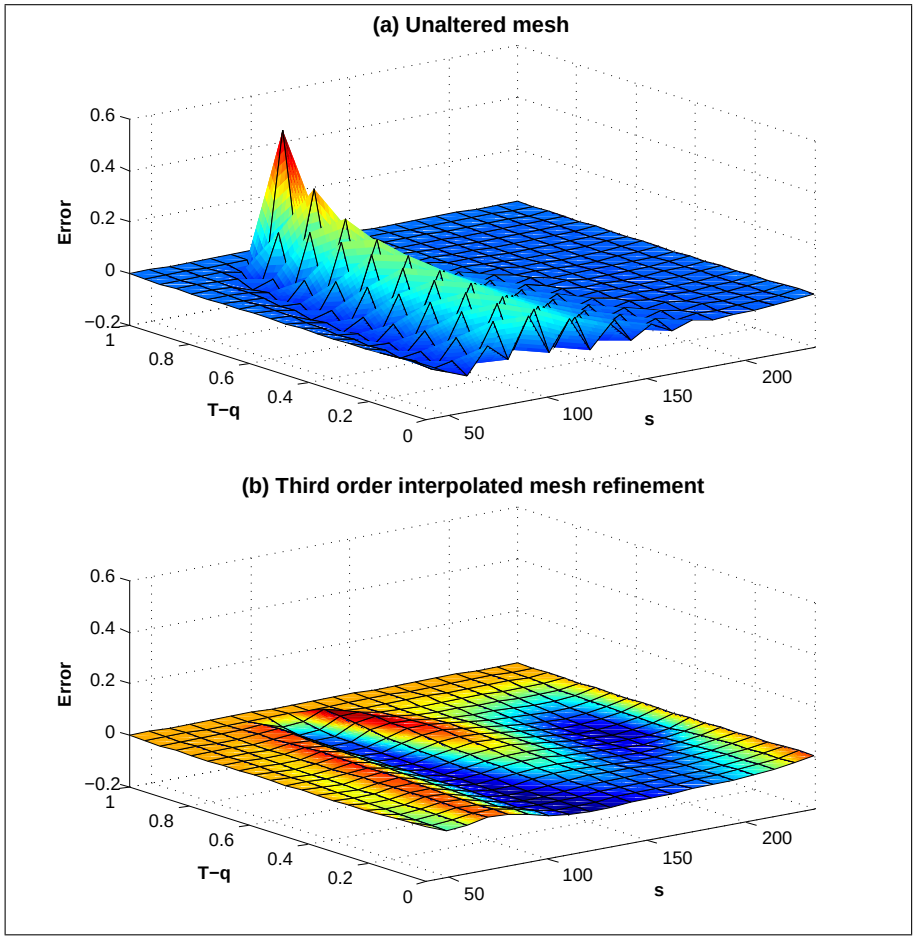

Figure 9.12: The difference between the Du Fort and Frankel scheme and the analytic solution for a European option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$. A third order interpolated grid refinement technique (b) is compared to an unaltered grid (a) with $M = 19$ and $N = 19$.

Method	**Effectiveness**	**Ease of implementation**	**Comment**
Analytical grid adjustment	Ineffective	Elegant and relatively easy to implement.	Appears to aggravate inconsistencies.
Temporal adjustment	Ineffective	Requires some additional code. Generally uncomplicated.	Appears to smooth errors rather than reducing them.
Classical adaptive mesh methods	Ineffective	Not possible	Three-time step structure prohibits implementation.
Linearly interpolated adaptive mesh	Ineffective	Difficult – requires many lines of additional code	Linear interpolation not sufficient.
Third order interpolated adaptive mesh	Promising	Difficult – requires many lines of additional code. Somewhat arbitrary.	Becomes spurious with finer meshes.

Table 9.1: A comparison between measures to improve performance at areas of steep gradients.

scheme utilizes two known time vectors in order to compute the unknown time vector. This detail complicates matters to a great extent.

We considered three classes of possible measures that may lead to improved results.

The first class is an analytical grid adjustment in the spatial direction. The adjustment is such that a concentration of grid points fall in the vicinity where numerical difficulty is expected. The solution on the transformed grid was transformed back to the original coordinates. Despite numerous reports in literature of improved results, our own efforts failed, possibly due to inconsistencies that arise as a result of an altered spatial-to-temporal step size on the adjusted grid.

A second measure considered was to concentrate the temporal steps near the maturity of the contingent claim where the greatest gradients in the solution are expected. The algorithm has to ensure that for any time step there are two equidistant time steps prior to it. The algorithm that was used achieved this, but the results did not improve significantly.

A third measure that was considered was to refine the grid in both the spatial and temporal direction around points where the expected error is the greatest. The expected error is assumed to have a functional relationship with the second spatial derivative. Difficulty arises due to the fact that the Du Fort and Frankel scheme uses two time steps to compute. This was overcome by using interpolations to fill in the grid points required for the calculation. Both second order linear and third order interpolations were used. The linear interpolations performs poorly as essential information is not captured by the interpolation technique. The third order interpolation technique performed well under certain conditions compared with an unaltered grid. It was found, however that the solution becomes spurious and possibly unstable if the spatial-to-temporal steps sizes are small. Generally the results are encouraging and further research into this matter may yield useful results.

Chapter 10

Free boundary value problems

10.1 Introduction

The purpose of this chapter is to establish the relative merits of the Du Fort and Frankel finite difference scheme, compared to the Crank and Nicolson scheme (as a representative of the class of implicit schemes) when free boundary value problems are encountered.

Free boundary problems are common in finance. A class of contingent claims that frequent the trading arena is the American option, where the holder has the right to exercise the option before maturity.

Free boundary value problems augment the upper and lower boundary conditions with the inequalities

$$F(s,q) \geq \begin{cases} \max(s-X,0) & \text{for call options, and} \\ \max(X-s,0) & \text{for put options,} \end{cases} \tag{10.1}$$

In the case of the American put option, we observe a critical "touching" point at spot price s_f. We observe

$$F(s,q) > \max(X-s,0) \quad \text{for } s > s_f,$$
$$F(s,q) = X - s \quad \text{for } s \leq s_f.$$

Therefore, for all underlying prices $s \leq s_f$, the rational holder of the option exercises the option, and we are therefore not required to calculate its price [Seydel, 2004]. The

location of s_f is not known *a priori* hence the term *free boundary*.

At the point s_f, two boundary conditions must hold for the American put option [Seydel, 2004], namely

$$\begin{aligned} F(s_f, q) &= X - s_f, \\ \frac{\partial F(s_f, q)}{\partial s} &= -1. \end{aligned} \tag{10.2}$$

These boundary conditions give rise to a partial differential inequality as opposed to the partial differential equation from equation (2.3) that we have solved so far. The inequality is given by

$$\frac{\partial F}{\partial q} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 F}{\partial s^2} + rs\frac{\partial F}{\partial s} - rF \leq 0. \tag{10.3}$$

The American option problem may be written as a linear complementarity problem: If $F(s,t) > \Phi(s,T)$ – the intrinsic value of the option – the function F is a solution of the Black and Scholes partial differential equation (equation (2.3)). If $F(s,t) = \Phi(s,T)$ then F is the solution of a strict partial differential inequality, similar to inequality (10.3).

In this document we are mainly concerned about the numerical estimation of the free boundary. Finite difference schemes have desirable properties relating to the calculation of contingent claims with free boundaries. This is due to the fact that we proceed from maturity, hence a value for the function $F(s,q)$ is generally known or is in the process of being calculated for each successive time step as we regress towards the inception date. The techniques utilized by implicit and explicit schemes differ with explicit schemes being simpler [Wilmott, 2000b] and computationally more efficient. This simplicity is a major driving factor in motivating a case in favor of explicit schemes, and in particular for the Du Fort and Frankel scheme. The remainder of this chapter focusses on the various techniques to numerically solve free boundary prolems, after which we compare results obtained from the Crank–Nicolson scheme and the Du Fort and Frankel scheme. Since no exact analytical solution exists, we assume the Crank and Nicolson scheme with $N = 1000$ and $M = 1000$ to be the benchmark for correctness.

10.2 American options and implicit finite difference methods

We solve an American put option using the Crank and Nicolson finite difference method. The the Crank–Nicolson difference equation is given by

$$A\hat{f}_{j+1}^{i+1} + (B-2)\hat{f}_j^{i+1} + C\hat{f}_{j-1}^{i+1} + A\hat{f}_{j+1}^{i} + (B+2)\hat{f}_j^{i} + C\hat{f}_{j-1}^{i} = 0.$$

In matrix from the Crank and Nicolson equation is given by equation (4.10), restated for convenience

$$\mathbf{M_L f^{i+1}} = \mathbf{M_R f^i} + \bar{\mathbf{b}}^{\mathbf{i}}, \qquad \bar{\mathbf{b}}^{\mathbf{i}} = \mathbf{b_R^i} - \mathbf{b_L^{i+1}}.$$

One technique of finding the free boundary is to compute $\mathbf{f^{i+1}}$ and test every element of the vector in order to determine whether equation (10.1) holds, and to manually adjust the values $F(s,t) < X - s$ to $F(s,t) = X - s$. Since the entire vector $\mathbf{f^{i+1}}$ is derived from the same system of equations, the implication is that every value of vector $\mathbf{f^{i+1}}$ is interlinked with every other value. By manually adjusting values after the vector was computed violates the matrix equality resulting in a local truncation error of $\mathcal{O}(k)$ rather than $\mathcal{O}(k^2)$ [Wilmott, 2000b]. The requirement is to replace the violating values at the same time as the values are found. A broad discussion of these methods falls outside the scope of this document, and are discussed in works related to this topic [for instance Wilmott, 2000b; Isaacson and Keller, 1966; Smith, 1984]. We briefly discuss the method we will be using, namely the successive over-relaxation (SOR) method, adopting arguments from Wilmott [2000b] and Smith [1984].

10.2.1 A brief discussion of the successive over-relaxation method

We rewrite matrix equation

$$\mathbf{M_L f^{i+1}} = \mathbf{M_R f^i} + \bar{\mathbf{b}}^{\mathbf{i}}$$

in the simpler form

$$\mathbf{Mv} = \mathbf{q}.$$

Assuming matrix M is square with N rows and columns and elements $M^i_j, \quad i,j = 1,2,\ldots,N$, then the set of equations

$$\begin{aligned} M^1_1 v_1 + M^1_2 v_2 + \ldots + M^1_N v_N &= q_1 \\ M^2_1 v_1 + M^2_2 v_2 + \ldots + M^2_N v_N &= q_2 \\ &\vdots \\ M^N_1 v_1 + M^N_2 v_2 + \ldots + M^N_N v_N &= q_N, \end{aligned}$$

are implied. These may be rewritten as

$$\begin{aligned} v_1 &= \frac{1}{M^1_1}\left(q_1 - (M^1_2 v_2 + M^1_3 v_3 + \ldots + M^1_N v_N)\right) \\ v_2 &= \frac{1}{M^2_2}\left(q_2 - (M^2_1 v_1 + M^2_3 v_3 + \ldots + M^2_N v_N)\right) \\ &\vdots \\ v_N &= \frac{1}{M^N_N}\left(q_N - (M^N_1 v_1 + M^N_2 v_2 + \ldots + M^N_{N-1} v_{N-1})\right). \end{aligned}$$

We iteratively approximate vector $\mathbf{v}$. We denote v^1_1 as the first approximation of element v_1, v^2_1 as the second approximation of v_1, v^n_1 as the n^{th} approximation and so on. The SOR method builds on the Jacobi method and the Gauss-Seidel method.

The Jacobi method approximates the entire vector $\mathbf{v}$ for the first iteration, and then uses this approximation in the second iteration, and so on. In general the Jacobi method may be written as

$$v^{n+1}_i = \frac{1}{M^i_i}\left\{ q_i - \sum_{j=1}^{i-1} M^i_j v^n_j - \sum_{j=i+1}^{N} M^i_j v^n_j \right\}, \qquad i = 1,2,\ldots,M. \tag{10.4}$$

The Gauss-Seidel method makes use of approximations as soon as they become available in order to speed up convergence, i.e.

$$v^{n+1}_i = \frac{1}{M^i_i}\left\{ q_i - \sum_{j=1}^{i-1} M^i_j v^{n+1}_j - \sum_{j=i+1}^{N} M^i_j v^n_j \right\}, \qquad i = 1,2,\ldots,M. \tag{10.5}$$

Adding and subtracting $v^n_i, \quad i = 1,2,\ldots,N$ to the right hand side of equation (10.5) we obtain

$$v^{n+1}_i = v^n_i + \frac{\omega}{M^i_i}\left\{ q_i - \sum_{j=1}^{i-1} M^i_j v^{n+1}_j - \sum_{j=i}^{N} M^i_j v^n_j \right\}, \quad i = 1,2,\ldots,M, \tag{10.6}$$

where for the moment $\omega = 1$. The right hand side terms in curly brackets are referred to as *displacements* as these terms are added to the previous iteration in order to obtain

a better approximation. Matrix $\mathbf{M}$ is such that successive corrections $v_i^{n+1} - v_i^n$ are all one-signed as n increases [Wilmott, 2000b]. The implication is that convergence may be speed up by choosing $\omega > 1$ and thereby assign a greater weight to the displacements. The optimal value of $1 < \omega < 2$ is when the spectral radius of the SOR matrix is a minimum. The SOR matrix given by [Wilmott, 2000b]

$$(\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{L})^{-1}((1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{U}),$$

where $\mathbf{I}$, $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{L}$ are the identity matrix, an upper triangular matrix, the diagonal matrix and a lower triangular matrix respectively such that $\mathbf{M} = \mathbf{U} + \mathbf{D} + \mathbf{L}$. We will not attempt to find the optimal value of ω, instead we make use of an algorithm described by Wilmott [2000b] that iteratively find a suitable value for ω. The idea of the algorithm is to start with $\omega = 1$ (Gauss-Seidel method) and to increase its value marginally after each iteration until the number of iterations fail to decrease.

```
function Q = SOR(M,q,v,tol, Payoff, omega)

MU = [diag(M,1);0];
ML = [0;diag(M,-1)];
MD = diag(M,0);
N = size(M,1);
NumberIterations = 0;
Error = tol+1;

temp = zeros(N,1);
while Error > tol
    Error = 0;
    for i = 1:N
        temp(i) = v(i+1)+omega*(q(i)-MU(i)*v(i+2)...
                                -MD(i)*v(i+1)...
                                -ML(i)*v(i))/MD(i);

        temp(i) = max(temp(i),Payoff(i+1));
        v(i+2) = max(v(i+2),Payoff(i+2));
```

```
        v(i) = max(v(i), Payoff(i));

        Error = Error + (temp(i) − v(i+1)) ∗ (temp(i)−v(i+1));
        v(i+1) = temp(i);
    end
    NumberIterations = NumberIterations + 1;
end
Q = {v, NumberIterations};
end
```

Listing 10.1: Matlab SOR function.

Listing (10.1) shows a Matlab implementation of a SOR algorithm, adapted from Wilmott [2000b].

10.3 The Du Fort and Frankel scheme for American options

Explicit schemes calculate the elements of the unknown temporal vector independent of each other. This property simplifies the computation of free boundary problems for explicit schemes and specifically the Du Fort and Frankel scheme. Wilmott [2000b] reports that by manually adjusting the calculated prices of claims in order to comply with arbitrage principles (recall that the price of an American option must equal at least its intrinsic value), preserves the conditions of equation (10.2). This leads to a simple implementation which only slightly differs from European options. The Du Fort and Frankel difference equation for an American option is given by[1]

$$f_j^{i+1} = \max\{\ddot{A}f_{j+1}^i + \ddot{B}f_j^i + \ddot{C}f_{j-1}^i + \ddot{D}f_j^{i-1}, X - s(j)\}.$$

10.4 Numerical results

The Du Fort and Frankel scheme and the Crank and Nicolson scheme performs virtually identically in pricing American put options. Figure (10.1) depicts the error between

[1] Compare with section (5.4.2)

a highly accurate Crank and Nicolson scheme with a Crank and Nicolson scheme with $N = 50$ and $M = 80$ (top) and a Du Fort and Frankel approximation with Richardsons extrapolation with $N = 50$, $M1 = 80$ and $N2 = 40$. Both schemes priced an American put option with $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$. From figure (10.1) it is apparent that the Du Fort and Frankel scheme performed virtually identical to the Crank and Nicolson scheme. The Du Fort and Frankel scheme with Richardson's extrapolation was roughly 35% quicker than the Crank and Nicolson scheme with a coarse grid. With 50 simulations the mean time it took the Du Fort and Frankel scheme with Richardson's extrapolation was 0.0107 seconds compared to 0.0144 seconds for the Crank and Nicolson algorithm.

For finer meshes, the Du Fort and Frankel scheme outperforms the Crank and Nicolson scheme. Figure (10.2) depicts the times to compute various mesh sizes. It is apparent that the Du Fort and Frankel scheme outperforms the Crank and Nicolson scheme for finer meshes.

10.5 Conclusion

The Du Fort and Frankel scheme and the Crank and Nicolson scheme produce similar American option prices. The Crank and Nicolson scheme makes use of a SOR algorithm that is computationally more expensive than the manual adjustment of grid points, employed by the Du Fort and Frankel scheme. We used Richardson's extrapolation with the Du Fort and Frankel algorithm, and for coarse meshes both the Crank and Nicolson and Du Fort and Frankel algorithms performed on par in terms of the time to compute. However, for finer grids, the Du Fort and Frankel scheme outperforms the Crank and Nicolson scheme.

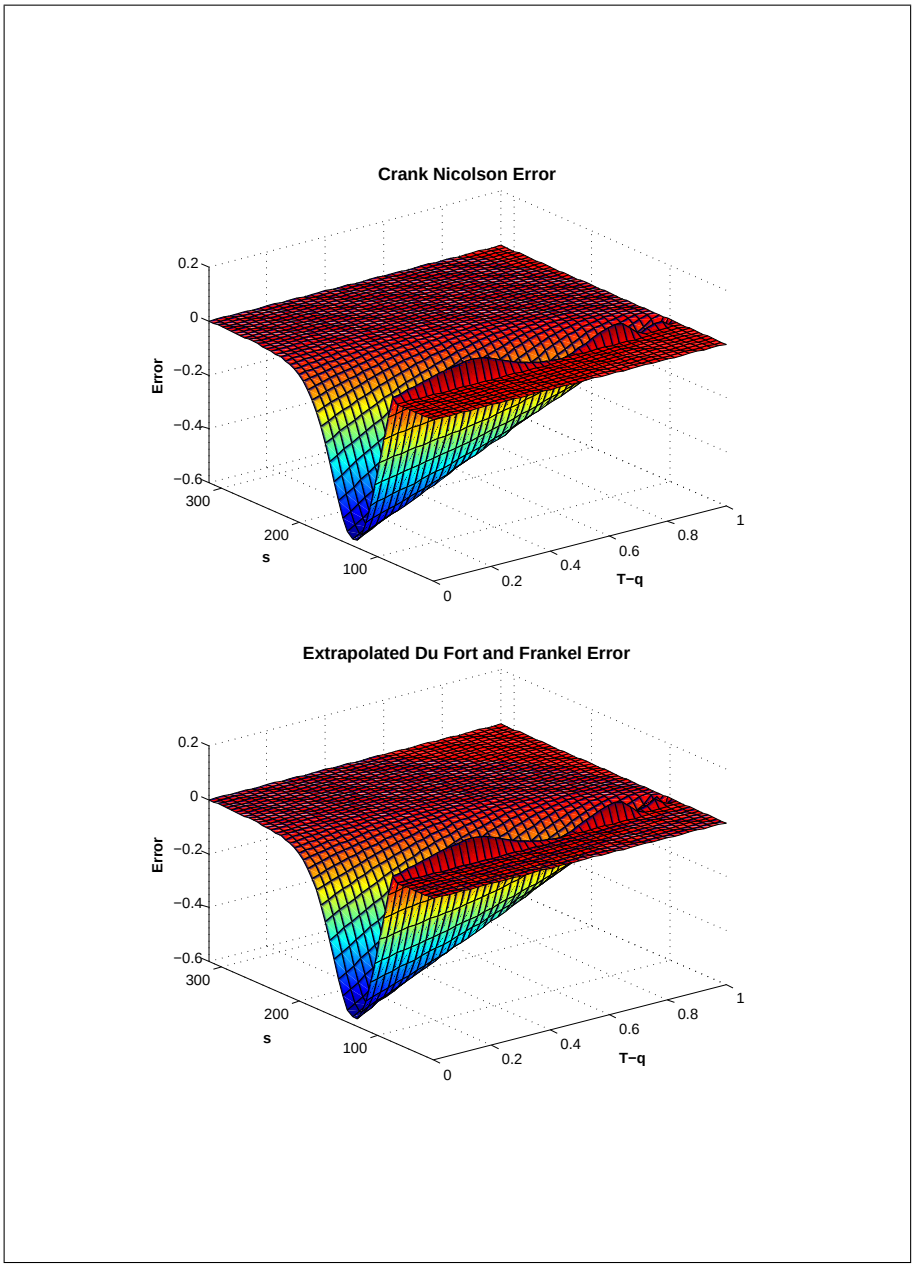

Figure 10.1: The error in pricing an American put option with $S_0 = 100$, $X = 100$, $T = 1$, $r = 0.1$ and $\sigma = 0.3$ using Crank and Nicolson scheme (top) with an $M = 80$ and $N = 50$ grid, and a Richardson extrapolated Du Fort and Frankel scheme (bottom) with $M1 = 80$, $M2 = 40$ and $N = 50$. The benchmark for correctness is a Crank and Nicolson approximation with $N = 1000$ and $M = 1000$.

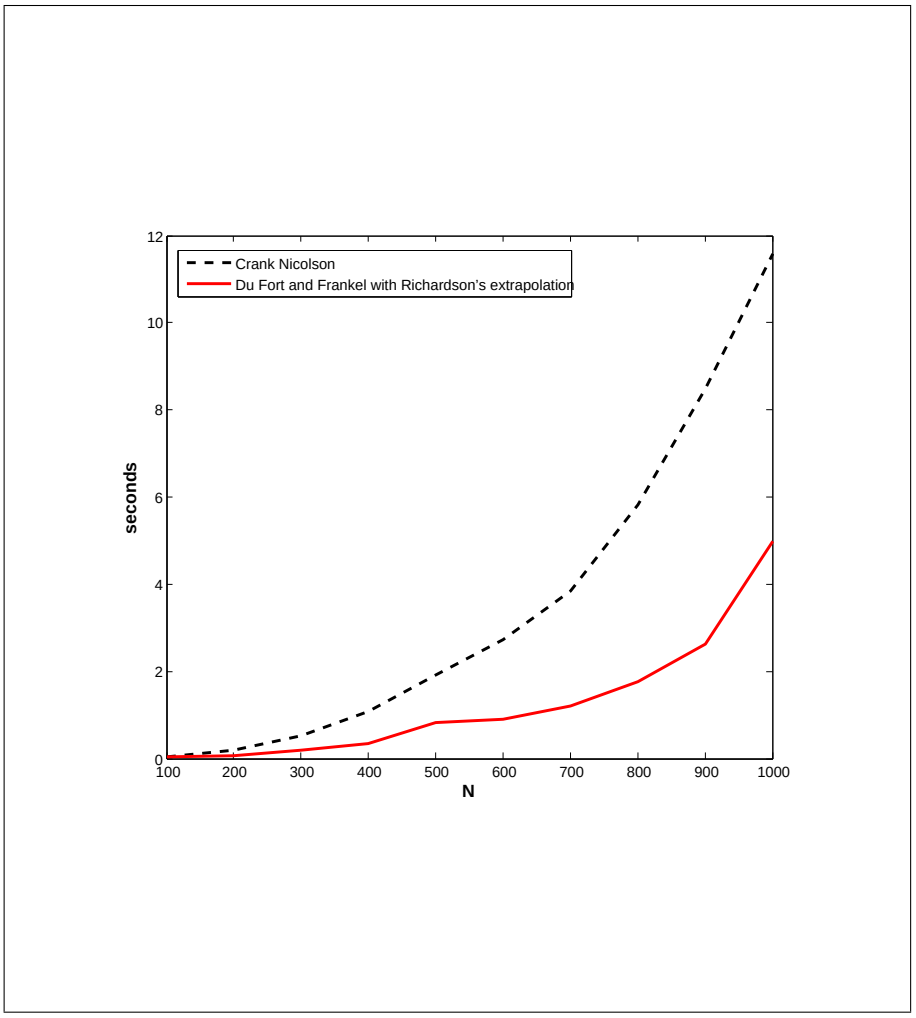

Figure 10.2: The time to compute an American put option with different grid sizes. For each value of N, the number of times steps are $M = 1.2N = M1$ with $M2 = \frac{1}{2}M1$ used for Richardson's extrapolation.

Chapter 11

Multi dimensional problems

11.1 Introduction

The purpose of this chapter is to establish whether the Du Fort and Frankel scheme is an efficient tool to price more dimensional contingent claims. Classically the finite difference method is not the method of choice for higher dimensions - such problems are more efficiently priced by making use of Monte Carlo methods. The reason for this is that the number of gird points that require computation grows exponentially, to the order of MN^d where d is the number of spatial dimensions. The number of Monte Carlo methods simulations grow linearly with the number of dimensions, and even if Monte Carlo methods are slow to converge in one dimension, it quickly overtakes finite difference schemes when dimensionality grows [Wilmott, 2000b].

A further complication with higher dimensionality relates to the inversion of matrices required by implicit schemes. This topic falls outside the scope of this document as the Du Fort and Frankel scheme, being explicit, escapes these complications.

Dimensionality not only grows with the addition of additional spatial variables, but generally with the addition of any stochastic variable, implying that the likes of Asian options (both the underlying share price and the average price are stochastic variables) and options with stochastic volatility all require additional dimensions. It is mainly for these low dimensional problems that finite difference methods are useful.

We will focus our attention on an option with a payoff function that is the maximum of

two assets minus a strike price,

$$\Phi(S_1(T), S_2(T), T) = \max(\max(S_1(T), S_2(T)) - X, 0),$$

also known as a variation of the class of *rainbow* options. We choose this option for two reasons: Firstly it has an analytic solution [Haug, 1998] which we can use as a benchmark, and secondly, this option falls within the Black and Scholes framework which we derived earlier.

11.2 Derivative discretisation

Our original derivation of the Black and Scholes partial differential equation (equation (2.3)) involved a spatial vector $S(t) = (S_1(t), S_2(t), \ldots)$ which up to now only consisted of a single element. We extend the vector by adding a second element. The partial differential equation may be rewritten as

$$\frac{\partial F}{\partial q} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 F}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 F}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 F}{\partial S_1 S_2} + rS_1 \frac{\partial F}{\partial S_1} + rS_2 \frac{\partial F}{\partial S_2} - rF = 0, \tag{11.1}$$

where ρ has the meaning described earlier (see equation 2.2). The other variables have their usual meaning.

Discretising the derivatives are identical to that described in section (5.4.2) but with an additional second derivative $\frac{\partial^2 F}{\partial S_1 S_2}$. We adopt the notation $f^i_{j,g} \equiv f(S_{1\chi} + jh_1, S_{2\chi} + gh_2, ik)$. Assuming there is a functional relationship between h_1 and h_2, i.e. $h_2 = xh_1$,

we consider the following Taylor expansions:

$$\begin{aligned}
\text{(a)}\quad f^i_{j+1,g+1} &= f^i_{jg} + h_1\frac{\partial f}{\partial s_1} + h_2\frac{\partial f}{\partial s_2} + \frac{1}{2}\left(h_1^2\frac{\partial^2 f}{\partial s_1^2} + 2h_2h_1\frac{\partial^2 f}{\partial s_1 s_2} + h_2^2\frac{\partial f}{\partial s_2^2}\right) + \dots \\
&\quad + \frac{1}{6}\left(h_1^3\frac{\partial^3 f}{\partial s_1^3} + 3h_1^2h_2\frac{\partial^3 f}{\partial s_1^2 s_2} + 3h_1h_2^2\frac{\partial^3 f}{\partial s_1 s_2^2} + h_2^3\frac{\partial^3 f}{\partial s_2^3}\right) + \dots \\
\text{(b)}\quad f^i_{j-1,g+1} &= f^i_{jg} - h_1\frac{\partial f}{\partial s_1} + h_2\frac{\partial f}{\partial s_2} + \frac{1}{2}\left(h_1^2\frac{\partial^2 f}{\partial s_1^2} - 2h_2h_1\frac{\partial^2 f}{\partial s_1 s_2} + h_2^2\frac{\partial f}{\partial s_2^2}\right) + \dots \\
&\quad + \frac{1}{6}\left(-h_1^3\frac{\partial^3 f}{\partial s_1^3} + 3h_1^2h_2\frac{\partial^3 f}{\partial s_1^2 s_2} - 3h_1h_2^2\frac{\partial^3 f}{\partial s_1 s_2^2} + h_2^3\frac{\partial^3 f}{\partial s_2^3}\right) + \dots \\
\text{(c)}\quad f^i_{j+1,g-1} &= f^i_{jg} + h_1\frac{\partial f}{\partial s_1} - h_2\frac{\partial f}{\partial s_2} + \frac{1}{2}\left(h_1^2\frac{\partial^2 f}{\partial s_1^2} - 2h_2h_1\frac{\partial^2 f}{\partial s_1 s_2} + h_2^2\frac{\partial f}{\partial s_2^2}\right) + \dots \\
&\quad + \frac{1}{6}\left(h_1^3\frac{\partial^3 f}{\partial s_1^3} - 3h_1^2h_2\frac{\partial^3 f}{\partial s_1^2 s_2} + 3h_1h_2^2\frac{\partial^3 f}{\partial s_1 s_2^2} - h_2^3\frac{\partial^3 f}{\partial s_2^3}\right) + \dots \\
\text{(d)}\quad f^i_{j-1,g-1} &= f^i_{jg} - h_1\frac{\partial f}{\partial s_1} - h_2\frac{\partial f}{\partial s_2} + \frac{1}{2}\left(h_1^2\frac{\partial^2 f}{\partial s_1^2} + 2h_2h_1\frac{\partial^2 f}{\partial s_1 s_2} + h_2^2\frac{\partial f}{\partial s_2^2}\right) + \dots \\
&\quad + \frac{1}{6}\left(-h_1^3\frac{\partial^3 f}{\partial s_1^3} - 3h_1^2h_2\frac{\partial^3 f}{\partial s_1^2 s_2} - 3h_1h_2^2\frac{\partial^3 f}{\partial s_1 s_2^2} - h_2^3\frac{\partial^3 f}{\partial s_2^3}\right) + \dots
\end{aligned}$$

From (a) - (b) - (c) + (d) we obtain

$$f^i_{j+1,g+1} - f^i_{j-1,g+1} - f^i_{j+1,g-1} + f^i_{j-1,g-1} = 4xh_1^2\frac{\partial^2 f}{\partial s_1 s_2} + \mathcal{O}(h_1^4)$$

$$\therefore \frac{\partial^2 f}{\partial s_1 s_2} = \frac{1}{4h_1h_2}(f^i_{j+1,g+1} - f^i_{j-1,g+1} - f^i_{j+1,g-1} + f^i_{j-1,g-1}) + \mathcal{O}(h_1^2). \quad (11.2)$$

The two dimensional Du Fort and Frankel scheme therefore exhibits a local truncation error of

$$T^i_{j,g} = \mathcal{O}(h_1^2, h_2^2, k^2, \frac{k^2}{h_1^2}, \frac{k^2}{h_2^2}).$$

The Du Fort and Frankel discretisation of equation (11.1) is found by substituting the relevant derivatives for their discretised versions

$$\begin{aligned}
&-\frac{1}{2k}(f^{i+1}_{j,g} - f^{i-1}_{j,g})\dots \\
+\frac{\sigma_1^2 s_1^2}{2h_1^2}(f^i_{j+1,g} - f^{i+1}_{j,g} - f^{i-1}_{j,g} + f^i_{j-1,g}) &+ \frac{\sigma_2^2 s_2^2}{2h_2^2}(f^i_{j,g+1} - f^{i+1}_{j,g} - f^{i-1}_{j,g} + f^i_{j,g-1})\dots \\
&+\frac{\rho\sigma_1\sigma_2 s_1 s_2}{4h_1h_2}(f^i_{j+1,g+1} - f^i_{j+1,g-1} - f^i_{j-1,g+1} + f^i_{j-1,g-1})\dots \\
&+\frac{rs_1}{2h_1}(f^i_{j+1,g} - f^i_{j-1,g}) + \frac{rs_2}{2h_2}(f^i_{j,g+1} - f^i_{j,g-1}) - rf = 0.
\end{aligned}$$

After rearranging terms we obtain

$$\begin{aligned} f_{j,g}^{i+1} &= A(f_{j+1,g+1}^{i} - f_{j+1,g-1}^{i} - f_{j-1,g+1}^{i} + f_{j-1,g-1}^{i}) \cdots \\ &\quad + Bf_{j+1,g}^{i} + Cf_{j,g+1}^{i} + Df_{j,g}^{i} + Ef_{j-1,g}^{i} \cdots \\ &\quad + Ff_{j,g-1}^{i} + Gf_{j,g}^{i-1}, \end{aligned} \tag{11.3}$$

where

$$\begin{aligned} A &= \frac{1}{2}H(h_1h_2k\rho\sigma_1\sigma_2s_1s_2), \\ B &= H(h_2^2k(\sigma_1^2s_1^2 + rh_1s_1), \\ C &= H(h_1^2k(\sigma_2^2s_2^2 + rh_2s_2), \\ D &= H(-2rh_1^2h_2^2k), \\ E &= H(h_2^2k(\sigma_1^2s_1^2 - rh_1s_1), \\ F &= H(h_1^2k(\sigma_2^2s_2^2 - rh_2s_2), \\ G &= H(h_1^2h_2^2 - h_2^2k\sigma_1^2s_1^2 - h1^2k\sigma_2^2s_2^2), \text{ and} \\ H &= (h_1^2h_2^2 + h_2^2k\sigma_1^2s_1^2 + h_1^2k\sigma_2^2s_2^2)^{-1}. \end{aligned}$$

11.3 Specification of boundaries

Boundary specification is not trivial in multiple dimensions. The terminal boundary is given by the payoff function, in the case of the option under consideration

$$f_{j,g}^{1} = \max\{\max(s_1(j), s_2(g)) - X, 0\}, \qquad j = 1, 2, \ldots, N_1{+}1, \quad g = 1, 2, \ldots, N_2{+}1.$$

Remaining are 4 boundaries, each consisting of a surface, rather than a line [Wilmott, 2000b], as is the case for one dimensional problems. These are

$$\begin{aligned} f_{1,g}^{i} &= \chi_0^1, \\ f_{N_1+1,g}^{i} &= \psi_0^1, \\ f_{j,1}^{i} &= \chi_0^2, \\ f_{j,N_2+1}^{i} &= \psi_0^2. \end{aligned}$$

For simplicity we assume these are known.

11.4 Performance of the Du Fort and Frankel scheme

The error of the Du Fort and Frankel compared to an analytical solution is depicted in figure (11.1). The speed performance of the algorithm is acceptable, but as expected

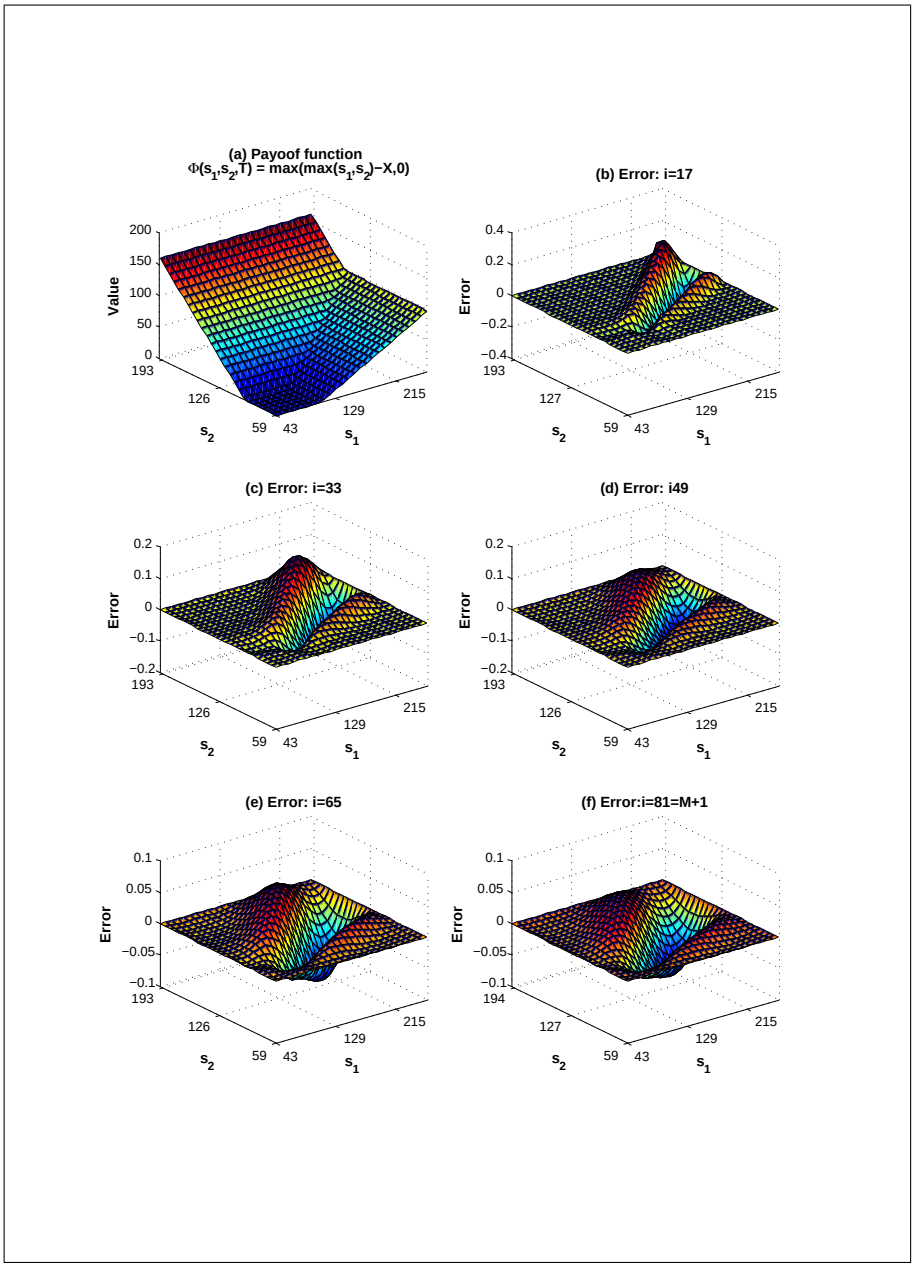

Figure 11.1: The payoff function (a) and the error in pricing a best-of-call Rainbow option with $X = 100$, $T = 1$, $r = 0.1$, $\sigma_1 = 0.3$, $\sigma_1 = 0.2$, and $\rho = 0.5$ using the Du Fort and Frankel scheme ($M = 80$, $N_1 = 50$, and $N_2 = 50$) compared to an analytical solution.

Grid size	Time (s)
$100 \times 100 \times 120$	0.3965
$200 \times 200 \times 240$	2.422
$300 \times 300 \times 360$	7.5088
$400 \times 400 \times 480$	20.39

Table 11.1: Time to compute different grid sizes by using the two dimensional Du Fort and Frankel scheme

grows quickly when the number of dimensions are increased. Table (11.1) shows the time it took for different mesh sizes to compute the price of a two-dimensional rainbow option. It is clear that additional accuracy comes at a high computational price.

11.5 Boundary free schemes

Hull and White [1990] observes that explicit schemes do not require upper and lower boundary conditions in order to achieve convergence. This property makes explicit schemes desirable from a computational point of view. The scheme becomes a quasi-trinomial tree. We experiment with the idea, hopeful that the stability characteristics of the Du Fort and Frankel scheme may alleviate some of the spatial-to-temporal steps size restrictions that are otherwise in place for explicit schemes. Since the number of spatial nodes grow by two with the addition of each time step the danger arises that the spatial variable may become negative if a sufficient spatial step size is used to ensure reasonable consistency properties. For this reason we log-transform the spatial dimensions, and consequently the partial differential equation. The resulting partial differential equation is [Tavella and Randall, 2000]

$$-\frac{\partial F}{\partial q}+\nu_1\frac{\partial F}{\partial S_1}+\nu_2\frac{\partial F}{\partial S_2}+\frac{1}{2}\sigma_1^2\frac{\partial^2 F}{\partial S_1^2}+\frac{1}{2}\sigma_2^2\frac{\partial^2 F}{\partial S_2^2}+\sigma_1\sigma_2\rho\frac{\partial^2 F}{\partial S_1 S_2}-rF=0,$$

where $\nu_x \equiv (r-\frac{1}{2}\sigma_x^2)$. From the partial differential equation, the coefficients from equation (11.3) changes appropriately.

The algorithm fixes $N_1 = N_2 = 2M$ at the boundary. Each time step back in time calculates a matrix of solutions $\mathbf{V^i}$ of size $j \times k, \quad j,k = 2(M-i)+3$. The effective number of nodes are halved, since the number of nodes grow with respect to dimension

with

$$W = M \prod_{i=1}^{d} N_d,$$

where W is the total number of grid points and d is the number of dimensions. Since we effectively half the number of grid points by leaving out the upper and lower boundary conditions, the total number of grid points are reduced to

$$\begin{aligned} \tilde{W} &= M \prod_{i=1}^{d} \frac{1}{2} N_d \\ &= \frac{W}{2^d}. \end{aligned}$$

A boundary free Du Fort and Frankel two dimensional algorithm is listed in Appendix (A.2.6).

11.5.1 Success with boundary free schemes

The algorithm listed in Appendix (A.2.6) markedly reduces the computational effort. An additional advantage of boundary free multi-dimensional schemes is that the upper and lower boundary conditions, which are often tedious to specify correctly, need not to be specified. An important observation by Hull and White [1990] is that errors originating from the boundaries are not present, thus higher accuracy may be obtained. Despite the promising properties of boundary free multidimensional schemes, we were unable to acceptable results with our algorithm.

11.6 Conclusion

Finite difference schemes are important tools in pricing problems with low dimensionality. The computational effort associated with additional dimensionality quickly becomes problematic and thus multi dimensional finite difference problems are often synonymous with efficiency problems.

We illustrated a two dimensional problem namely a 2 colour rainbow call option, and found that the Du Fort and Frankel algorithm performed well. It was observed that the time to compute grows exponentially with the addition of additional grid points. We experimented with a technique whereby the upper and lower boundaries are left out in order to speed up computation.

Chapter 12

Part II conclusion and summary

In Part II we exposed the Du Fort and Frankel to a number of numerical challenges that frequently occur in finance. In Part I the scheme was found to have desirable properties on the basis of theoretical analysis.
We exposed the Du Fort and Frankel scheme to four classes of financial problems. Although many more potential problems exist, we believe that the general suitability of the Du Fort and Frankel scheme pertaining to finance will suitably be tested. The four problems were:

- Chapter 8 dealt with the presence of dividends causing discontinuous jumps,
- Chapter 9 provide a treatise on the presence of steeps gradients and singularities,
- Chapter 10 discussed free boundary problems, and
- Chapter 11 dealt with more dimensional problems.

12.1 Dividends

We found that the Du Fort and Frankel scheme copes well with discrete dividends, but not better than alternative schemes. The adjustment to pricing algorithms in order to incorporate dividends are in fact more problematic for three time-step schemes such as the Du Fort and Frankel schemes than for conventional schemes.

12.2 Discontinuous behavior

A number of measures were investigated in order to improve the accuracy of the Du Fort and Frankel scheme at areas where discontinuities or steep gradients in the solution occur. Most of our attempts were unsuccessful despite numerous reports in literature of success obtained for other schemes. Two characteristics of the Du Fort and Frankel scheme may collaborate in this absence of success. These are the fragile consistency properties of the scheme and the fact that it is a three time-step method which complicates grid refinements.
We believe that the Du Fort and Frankel scheme is not the most suitable scheme where contingent claims with a high degree of discontinuity – such as barrier options – are to to be priced.

12.3 Free boundary options

The Du Fort and Frankel scheme with Richardson's extrapolation was found to be an effective tool to price options with free boundaries, compared to implicit schemes, in particular the Crank and Nicolson scheme. The Du Fort and Frankel scheme is explicit, thus it is not required to iteratively find solutions to matrix equations as is the case with implicit schemes. This vastly improves the efficiency of the scheme and it was found that the benefit of using the Du Fort and Frankel scheme increases with required accuracy.

12.4 Multi-dimensional problems

For low dimensional problems, the Du Fort and Frankel scheme may prove useful. It's utility may even be more profound if boundary free schemes can be implemented with success. Our experimentation with boundary free schemes did not provide acceptable results. The use of the finite difference method for multi-dimensional problems remains limited.

12.5 Summary

Table 12.1 provides a summary of the suitability of the Du Fort and Frankel scheme for various problems in finance.

Problem	**Suitability**	**Ease of implementation**	**Comment**
Dividends	Useful	Relatively simple, but more complex compared to two time-step methods.	
Discontinuous behavior	Not suitable	Difficult.	Consistency problems and problems with multiple time-steps renders the scheme unpractical.
Free boundary problems	Effective	Simple	Tests were conducted with Richardson's extrapolation.
Multidimensional problems	Limited used	Relatively simple	

Table 12.1: Summary of the suitability of the Du Fort and Frankel scheme when encountering various problems in finance.

Part III

Conclusion

Chapter 13

Further research

13.1 Introduction

Throughout the research of topics pertaining to this document we identified areas where further research is required. This requirement stems from three sources. Firstly, there were a number of instances where we were unable to replicate results obtained by other authors. A second source prompting new research originates from a general lack specific research on the Du Fort and Frankel scheme, and especially for linear second order parabolic partial differential equations. The third source of required research stems from our own experimentation.

13.2 Replication of results

13.2.1 Stability testing for Black and Scholes equation

In Chapter 4 we derived stability conditions for the fully explicit scheme for the initial value Black and Scholes partial differential equation. The only other source to our knowledge that conducts this derivation was done by Wilmott [2000b]. We were unable to precisely replicate the results obtained by Wilmott [2000b].

In Chapter 5 we attempted to prove stability for the Du Fort and Frankel scheme for the Black and Scholes equation. The Du Fort and Frankel scheme is known to be unconditionally stable for the heat equation Smith [1984]. We were unable to source any analysis on partial differential equations with convection terms, and our own analysis

is not exhaustive. Gottlieb and Gustafsson [1976] makes an observation that stability of the Du Fort and Frankel scheme is not clear when coefficients are not constant. This is of special interest to the pricing of contingent claims as coefficients are often not constant.

13.2.2 Douglas schemes for PDE's containing convection terms

A statement by Wilmott [2000b] implies that Douglas schemes can be derived for the Black and Scholes partial differential equation. This is in conflict with our own finding, which is also supported by Smith [1984], namely that only second derivatives allow for the elimination of fourth order differences. We were unable to find any further research on this topic.

13.2.3 Spurious behavior due to central differencing and time averaging

In Chapter 6 we were unable to replicate results by Duffy [2004b] stating that time averaging and central differencing are responsible for spurious behavior in the Crank and Nicolson scheme.

13.2.4 Analytical grid refinement

In Chapter 9 we were unable to replicate the accuracy improvements from analytical grid adjustments reported by Oosterlee et al. [2004], Leentvaar and Oosterlee [2006] and Clarke and Parrot [1999] amongst others. A similar inability to replicate accuracy improvements was reported by Sottoriva and Rexhepi [2007] but no analysis of the problem was conducted. We speculated on one possibility namely that the concentration of mesh points in the spatial direction is causing a too low spatial–to–temporal step size ratio and consequently inconsistencies occur. Our initial analysis supports this notion to the extent that it was shown that inconsistencies do occur, but we were unable to determine any firm causality. Tavella and Randall [2000] cautions briefly against the use of analytical mesh refinement, but no analysis is shown. Research may not only isolate the cause of poor performance of the Du Fort and Frankel scheme in this regard, but in addition may also offer a remedy.

13.3 General research required

13.3.1 The impact of inconsistency

In Chapter 6 we obtained promising results making use of Richardson's extrapolation in order to reduce the impact of inconsistency on results obtained by the Du Fort and Frankel scheme. Further research may be required in order to make similar derivations for different partial differential equations.

13.3.2 Discrete dividends

The works of Beneder and Vorst [2001] and Bos and Vandermark [2002] were discussed in Chapter 8. These techniques provide analytical estimations for options where discrete dividends are present. From experimentation, both methods performed exceptional, however Haug et al. [2003] warns that no theoretical base for these models exist, and that using it may be risky. Proper testing of these techniques may prove fruitful, and a numerical treatise for discrete dividends may only be required in the presence of path dependent contingent claims.

13.3.3 Adaptive mesh techniques

In Chapter 9 the topic of adaptive mesh methods were investigated. The three time-step nature of the Du Fort and Frankel scheme hinders simple implementation, and we resorted to interpolation methods to find absent mesh point values. We utilized both linear interpolation as well as a third order interpolation technique. The third order interpolation technique returned encouraging results when a coarse grid was used. However, when finer grids were used, the method became spurious for no apparent reason. Research is required in order to isolate the reason for spurious behavior.

In general, more research is required in relation to the performance of the Du Fort and Frankel scheme where steep gradients in the solution occurs. We were unable to generate any acceptable results by adapting the conventional techniques for the Du Fort and Frankel scheme.

13.3.4 Boundary free schemes

In Chapter 11 we experimented with boundary free schemes. In a paper by Hull and White [1990] it was reported that explicit schemes do not require upper and lower boundary conditions and as a result such schemes have the potential to become more accurate as errors stemming from the boundaries do not form part of the solution. Boundary free schemes offer a compelling case in favor of explicit schemes, yet very little research after Hull and White [1990] was found.

We were unable to achieve acceptable results in terms of our own experimentation with a boundary free Du Fort and Frankel scheme. Successful implementation of a boundary free scheme may yield spectacular results in two respects. Firstly, boundary specification for multidimensional problems are often problematic, and boundary free schemes trivially takes care of this. Secondly, by halving the number of mesh points that require calculation, higher dimensional problems may become viable, or alternatively higher accuracy can be obtained.

Chapter 14

Conclusion

Conclusion

The Du Fort and Frankel finite difference scheme was applied to a class of financial problems. The document was structured in such a way as to systematically derive the Du Fort and Frankel scheme from the underlying financial problem, and then apply the scheme to obstacles in finance that frequently occur.

The first part of the document derived the Black and Scholes partial differential equation and adapted it to an initial value equation. It was also adapted to the heat equation as this simpler version is often used in literature to conduct analysis.

In the subsequent chapter we derived the three classical finite difference schemes, namely the implicit, explict and Crank and Nicolson schemes. These were evaluated and compared on the basis of truncation error, consistency and stability.

The analysis conducted up to Chapter 4 served as a foundation on which to conduct analysis on the main subject of this document namely the Du Fort and Frankel scheme. As a prelude to this scheme we analysed schemes that share important properties of the Du Fort and Frankel scheme, namely a second order spatial truncation error and an explicit scheme that is also stable. We chose the Richardson scheme to illustrate the second order temporal truncation error, and the MADE scheme, which share some stability characteristics with the Du Fort and Frankel scheme.

A consequence of the stability of the Du Fort and Frankel scheme is inconsistency with the partial differential equation, which was studied in the next chapter, together

with spurious oscillations ascribed to dominance of convection terms in partial differential equations. We found that consistency problems with the Du Fort and Frankel scheme can effectively be reduced by firstly increasing the number of temporal steps, and secondly by eliminating high order inconsistent error terms with Richardson's extrapolation. The second measure was found to be especially effective.

The second part of the document applied the Du Fort and Frankel scheme to financial problems. We focussed on four phenomena that frequents the financial arena. These were the presence of jumps due to dividend payments, singularities and discontinuous behavior, free boundary value problems and multi dimensionality.

We found that finite difference schemes present an effective means to price contingent claims of financial instruments with dividend payments. The conventional treatment required by two time-step schemes was amended in order to work for three time-step methods such as the Du Fort and Frankel scheme.

We studied various techniques in order to effectively price contingent claims where steeps gradients are present. We tested various ideas including analytical mesh refinement on the spatial dimension, temporal grid refinement and local grid refinement by the insertion of additional grid points at problematic areas. We experienced difficulty with all the techniques due to a number of reasons. These include the poor consistency tendency, and the requirement of multiple time steps in order to compute a solution of the Du Fort and Frankel scheme. We derived a method where grid refinement is achieved with third order interpolations. The new method had promising features, but displayed some undesirable characteristics which requires further research.

The Du Fort and Frankel scheme was found to be an efficient tool to compute American options. It was compared to the Crank and Nicolson scheme and it was found that for fine meshes the Du Fort and Frankel scheme computes results more efficiently.

We applied the Du Fort and Frankel scheme to a more dimensional problem and it was found that it performed well both in terms of accuracy and in terms of computing time. In order to speed up convergence further, we experimented with an idea to remove the upper and lower boundary conditions, thereby halving the number of grid points for each dimension. We were unable to achieve acceptable results with boundary free methods.

The condensed conclusion of this document is that the main theoretical shortcomings of the Du Fort and Frankel scheme can be effectively minimized and with proper pre-

caution scheme may prove to be a valuable addition to the suite of numerical techniques used for the pricing of contingent claims. The practical application of the scheme is limited where a high degree of discontinuity occurs, such as barrier options. Promising results were obtained for free boundary problems. The scheme also proves suitable for instances where discrete dividends occur and with general problems with low dimensionality.

Part IV

Appendices

Appendix A

Source code for miscellaneous functions

A.1 Analytical functions

A.1.1 Function to calculate the cumulative normal density function

```
function Q = CND(x)
%Standardized Cumulative Normal Distribution function
y = x/sqrt(2);
Q = double((erf(y)+1)/2);
```

A.1.2 Function to calculate the value of the genralized Black and Scholes formula

```
function Q = bsmatrix(PorC, S,X,T,r,b,v)

%*************************************************************************
%            GENRALIZED BLACK AND SCHOLES MATRIX ANALYTICAL MODEL
%
% Abie Bouwer
```

```
% November 2007
%
% Algorithm adapted from Haug, E. "The complete guide to option pricing
%          formulas", McGraw-Hill, 1997
%
%========================================================================
%Returns a matrix over S and T. Default test values are:
%PorC = 'P';          %Call or Put
%S = 50:10:200;       %Spot price range
%X = 100;             %Strike price
%T = 0:0.1:10;        %Time range
%r = 0.07;            %Risk free interest rate
%b = 0.07;            %Carry cost
%v = 0.25;            % Volatility
%========================================================================

TT = max(T,eps);
SS= max(S,eps);
[s,t] = meshgrid(SS,TT);
d1 = (log(s/X)+(b+0.5*(v*v))*t )./( v*sqrt(t ));
d2 = d1 - v*sqrt(t );
if PorC == 'C'
    Q = (s.* exp((b-r).*t )).* CND(d1)-(X * exp(-r * t)).* CND(d2);
else
    Q = X * exp(-r.*t ).* CND(-d2) - s.* exp((b-r)*t).* CND(-d1);
end
```

A.1.3 Function to calculate the analytical value of a barrier option

```
function Q = barriermatrix (PorC, S,X,T,r,b,v,K,H,InOut, UpDown)

%========================================================================
%                    BARRIER MATRIX ANALYTICAL MODEL
```

```
%
% Abie Bouwer
% November 2007
%
% Algorithm adapted from Haug, E. "The complete guide to option pricing
%          formulas", McGraw-Hill, 1997
%
%========================================================================
%Returns a matrix over S and T. Default test values are:
% PorC = 'C';    % Call or put
% K=0;           % Rebate
% H = 200;       % Barrier
% S = 50:10:200;% Spot range
% X = 100;       % Strike price
% T = 0:0.1:1;   % Time to maturity range
% r = 0.1;       % Risk free rate
% b = 0.1;       % Cost of carry
% v = 0.25;      % Volatility
% InOut = 'O';   % 'In' barrier or 'Out' barrier
% UpDown = 'U'; % 'Up' barrier or 'Down' barrier
%========================================================================

TypeFlag = {[PorC, UpDown, InOut]};
TT = max(T,eps);
SS= max(S,eps);
[s,t] = meshgrid(SS,TT);


mu = (b - v ^ 2 / 2) / v ^ 2;
lambda = sqrt(mu ^ 2 + 2 * r / v ^ 2);
X1 = log(s / X) ./ (v * sqrt(t)) + (1 + mu) * v * sqrt(t);
X2 = log(s / H) ./ (v * sqrt(t)) + (1 + mu) * v * sqrt(t);
y1 = log(H ^ 2 ./ (s * X)) ./ (v * sqrt(t)) + (1 + mu) * v * sqrt(t);
```

```
y2 = log(H ./ s) ./ (v * sqrt(t)) + (1 + mu) * v * sqrt(t);
Z = log(H ./ s) ./ (v * sqrt(t)) + lambda * v * sqrt(t);

if strcmp(TypeFlag,'CDI') || strcmp(TypeFlag, 'CDO')
    eta = 1;
    phi = 1;
    elseif strcmp(TypeFlag, 'CUI') || strcmp(TypeFlag,'CUO')
        eta = -1;
        phi = 1;
        elseif strcmp(TypeFlag,'PDI') || strcmp(TypeFlag,'PDO')
            eta = 1;
            phi = -1;
            elseif strcmp(TypeFlag,'PUI') || strcmp(TypeFlag,'PUO')
                eta = -1;
                phi = -1;
end

f1 = phi*s.*exp((b-r).* t ).*CND(phi*X1) - ...
    phi*X*exp(-r*t).*CND(phi*X1-phi*v*sqrt(t));
f2 = phi*s.*exp((b-r).*t ).*CND(phi*X2) - ...
    phi*X*exp(-r*t).*CND(phi*X2-phi*v*sqrt(t));
f3 = phi*s.*exp((b-r).*t ).*( H./s ).^(2*( mu+1)).*CND(eta*y1) - ...
    phi*X.*exp(-r*t).*(H./s ).^(2* mu).*CND(eta*y1-eta*v*sqrt(t));
f4 = phi*s.*exp((b-r).*t ).*( H./s ).^(2*( mu+1)).*CND(eta*y2) - ...
    phi*X.*exp(-r*t).*(H./s ).^(2* mu).*CND(eta*y2-eta*v* sqrt(t));
f5 = K*exp(-r*t).*(CND(eta*X2-eta*v*sqrt(t)) - ...
    (H./s ).^(2* mu).*CND(eta*y2-eta*v*sqrt(t)));
f6 = K*((H./s ).^( mu+lambda).*CND(eta*Z) + ...
    (H./s ).^( mu-lambda).*CND(eta*Z-2*eta*lambda*v*sqrt(t)));

TypeFlag = [PorC, UpDown, InOut];
if X > H
    switch TypeFlag
```

```
        case ('CDI')
            Q = f3 + f5;
        case ('CUI')
            Q = f1 + f5;
        case ('PDI')
            Q = f2 - f3 + f4 + f5;
        case ('PUI')
            Q = f1 - f2 + f4 + f5;
        case ('CDO')
            Q = f1 - f3 + f6;
        case ('CUO')
            Q = f6;
        case ('PDO')
            Q = f1 - f2 + f3 - f4 + f6;
        case ('PUO')
            Q = f2 - f4 + f6;
    end
elseif X < H
    switch TypeFlag
        case ('CDI')
            Q = f1 - f2 + f4 + f5;
        case ('CUI')
            Q = f2 - f3 + f4 + f5;
        case ('PDI')
            Q = f1 + f5;
        case ('PUI')
            Q = f3 + f5;
        case ('CDO')
            Q = f2 + f6 - f4;
        case ('CUO')
            Q = f1 - f2 + f3 - f4 + f6;
        case ('PDO')
            Q = f6;
```

```
        case ('PUO')
            Q = f1 - f3 + f6;
    end
end
```

A.2 Finite difference algorithms

A.2.1 The classical suite: Explicit, Crank and Nicolson and Explicit schemes

```
%*************************************************************************
%                    GENERALIZED FINITE DIFFERENCE ALGORITHM
%
% Abie Bouwer
%
% November 2007
%
%*************************************************************************
%
%The algorithm calculates the value of an option using a generalized finite
%difference scheme. Adjusting the paramater 'O' sets the convention:
%   O = 0.0 ( Explicit scheme)
%   O = 0.5 (Crank--Nicolson scheme)
%   O = 1.0 ( Implicit scheme)
%
%The payoff is specified in the section "BOUNDARY CONDITIONS"
%
%*************************************************************************

clear;
clc;

%INPUT PARAMETERS
```

```
%*************************************************************************
T = 1;              %Years to maturity
s0 = 100;           %Initial Spot
r = 0.15;           %Riskfree Interest Rate
o = 0.01;           % Volatility

M = 200;            %Number of time steps
N = 50;             %Number of spatial steps
O = 0.5;            %O = 0 for explicit , 0.5 for CN and 1 for implicit
X = 100;            %Strike price
%*************************************************************************

%GRID INITIALIZATION
%*************************************************************************
smin = 0;
smax = s0*exp((r-0.5*o*o)*T+3*o*sqrt(T));

k = T/M;            %Time step spacing
h = (smax-smin)/N; %Spatial step spacing

s = smin:h:smax;    %Spot prices
q = T:-k:0;         %Times

V =zeros(N+1,M+1); %Option price initialization
%*************************************************************************

%*************************************************************************
%BOUNDARY CONDITIONS
%
%The boundary conditions should reflect the payoff of the option
%*************************************************************************
%European Option Dirichlet Boundary conditions
V(1:N+1,1:M+1) = zeros(N+1,M+1);
```

```
V(:,1) = max(s-X,0);
V(1,:) = 0;
V(N+1,:) = smax-X*exp(r*(q-T));
%*************************************************************************

%FD variables
alpha = 0.5*o.^2*s.^2;
beta = r.*s;
gamma = -r;

A = (alpha*k)/h^2 + (beta*k)/(2*h);
B = gamma*k - (2*alpha*k)/(h^2);
C = (alpha*k)/h^2 - (beta*k)/(2*h);

%Lefthand Matrix...
ML = diag((1-O*B(2:N)),0) + ...
     diag((-O*A(2:(N-1))),1) + ...
     diag((-O*C(3:N)),-1);

%Righthand Matrix...
MR = diag((1+(1-O)*B(2:N)),0) + ...
     diag(((1-O)*A(2:(N-1))),1) + ...
     diag(((1-O)*C(3:N)),-1);

%Left boundary conditions ...
bL = zeros(N-1,M+1);
bL(1,:) = -O*C(2)*V(1,:);
bL((N-1),:) = -O*A(N)*V((N+1),:);

%Right boundary conditions ...
bR = zeros(N-1,M+1);
bR(1,:) = (1-O)*C(2)*V(1,:);
bR((N-1),:) = (1-O)*A(N)*V((N+1),:);
```

```
%*************************************************************************
%MAIN CALCULATION
%*************************************************************************
for i = 2:(M+1)
    V(2:N,i) = ML\((MR*V(2:N,i-1))+bR(:,(i-1))-bL(:,i));
end

Value = interp1(s,V(:,M+1),s0,' spline ');
```

A.2.2 The MADE scheme

```
%*************************************************************************
%                    MADE FINITE DIFFERENCE ALGORITHM
%
% Abie Bouwer
%
% November 2007
%
%*************************************************************************
%
%The algorithm calculates the value of an option using the MADE finite
%difference scheme.
%
%The payoff is specified in the section "BOUNDARY CONDITIONS"
%
%*************************************************************************

clear;
clc;


%INPUT PARAMETERS
```

```
%*************************************************************************
S0 = 100;    % Initial spot price
X = 100;     %Strike price
sig = 0.25; % Volatility
T = 1;       %Time to maturity
r = 0.1;     %Risk free interest rate

N = 20;
M = 440;
%*************************************************************************

%GRID INITIALIZATION
%*************************************************************************
Smax = S0*exp((r-0.5*sig*sig)*T+3*sig*sqrt(T));
Smin = 0;

h = (Smax-Smin)/N; %Spatial step size
k = T/M;           %Tempooral step size

s = Smin:h:Smax;   %Spatial vector
t = T:-k:0;        %temporal vector

U = zeros(N+1,M+1); %solution matrix initialization
%*************************************************************************

%*************************************************************************
%BOUNDARY CONDITIONS
%
%The boundary conditions should reflect the payoff of the option
%*************************************************************************
%European Option Dirichlet Boundary conditions
U(:,1) = max(s-X,0);
U(N+1,:) = Smax - X*exp(-r*(T-t));
```

```
U(1,:) = 0;

% FD variables
alpha = 0.5*sig.^2*s.^2;
beta = r.*s;
gamma =-r;

A = (2*alpha*k+beta*k*h)./(2*h^2+4*alpha*k);
B = (h^2+gamma*k*h^2)./(h^2+2*alpha*k);
C = (2*alpha*k-beta*k*h)./(2*h^2+4*alpha*k);

%We put the solution as f(i+1) = M*f(i)+ b ...

MR = diag(B(2:N),0)+diag(A(2:(N-1)),1)+diag(C(3:N),-1);
b = zeros(N-1,M+1);
b(1,:) = C(2)*U(1,:);
b((N-1),:) = A(N)*U((N+1),:);



%************************************************************************
%MAIN CALCULATION
%************************************************************************

for i = 2:(M+1)
    U(2:N,i) = (MR * U(2:N,i-1))+b(:,(i-1));
end
```

A.2.3 The standard Du Fort and Frankel Scheme

```
%************************************************************************
%              DU FORT AND FRANKEL FINITE DIFFERENCE ALGORITHM
%
% Abie Bouwer
```

```
%
% November 2007
%
%************************************************************************
%
%The algorithm calculates the value of an option using the Du Fort and
%Frankel finite difference scheme.
%
%The payoff is specified in the section "BOUNDARY CONDITIONS"
%
%************************************************************************

clear;
clc;

%INPUT PARAMETERS
%************************************************************************
S0 = 100;     % Initial spot price
X = 100;      %Strike price
T = 1.0;      %Time to maturity
r = 0.15;     %Risk free interest rate
o = 0.3;      % Volatility

N = 50;       %Spatial steps
M = 200;      %Temporal steps
%************************************************************************

%GRID INITIALIZATION
%************************************************************************
k = T/M;                  %Temporal step size
smax = S0*exp((r-0.5*o*o)*T+3*o*sqrt(T));
smin = 0;
h = (smax-smin)/N; %Spatial step size
```

```
s = smin:h:smax;      %Spot prices
q = T:-k:0;           %Times

V = zeros(N+1,M+1); %Solution matrix  initialization
%*************************************************************************

%*************************************************************************
%BOUNDARY CONDITIONS
%
%The boundary conditions should reflect the payoff of the option
%*************************************************************************
%European Option Dirichlet Boundary conditions
V(:,1) = max(s-X,0);
V(1,:) = 0;
V(N+1,:) = smax - X* exp(r* (q-T));
%*************************************************************************

%FD Variables
alpha = 0.5*o^2*s .^2;
beta = r*s;
gamma = -r;

%FIRST STEP WITH ANALYTIC FORMULA
V(:,2) = flipdim(bsmatrix('C', s,100,k,r,r,o),1)';

%Remaining Time steps with Du Fort and Frankel
A = (2*alpha*k + beta*k*h)./(h^2 + 2*alpha*k);
B = (2*gamma*k*h^2)./(h^2 + 2*alpha*k);
C = (2*alpha*k - beta*k*h)./(h^2 + 2*alpha*k);
D = (h^2 - 2*k*alpha)./(h^2 + 2*alpha*k);

MatrixM = diag(B(2:N),0) + ...
```

```
    diag(A(2:N-1),1) + ...
    diag(C(3:N),-1);
MatrixBoundary = zeros(N-1,M+1);
MatrixBoundary(1,:) = C(2)*V(1,1:M+1);
MatrixBoundary(N-1,:) = A(N)*V(N+1,1:M+1);

for i = 3:M+1
    V(2:N,i)= MatrixM*V(2:N,i-1) + MatrixBoundary(1:N-1,i-1) + D(2:N)'.*V(2:N,i-2);
end

Premium = interp1(s(1:N+1),V(1:N+1,M+1),S0,'linear');
```

A.2.4 The Du Fort and Frankel scheme with one-sided convection

```
%*************************************************************************
%  ONE SIDED CONVECTION DU FORT AND FRANKEL FINITE DIFFERENCE ALGORITHM
%
% Abie Bouwer
%
% November 2007
%
%*************************************************************************
%
%The algorithm calculates the value of an option using the Du Fort and
%Frankel finite difference scheme where the convection term makes use of
%one-sided differences instead of central differencing .
%
%The payoff is specified in the section "BOUNDARY CONDITIONS"
%
%*************************************************************************

clear;
clc;
```

```
%INPUT PARAMETERS
%*************************************************************************
S0 = 100;    % Initial spot price
X = 100;     %Strike price
T = 1.0;     %Time to maturity
r = 0.15;    %Risk free interest rate
o = 0.01;     % Volatility

N = 50;      %Spatial steps
M = 200;     %Temporal steps
%*************************************************************************

%GRID INITIALIZATION
%*************************************************************************
k = T/M;                %Temporal step size
smax = S0*exp((r-0.5*o*o)*T+3*o*sqrt(T));
smin = 0;
h = (smax-smin)/N; %Spatial step size

s = smin:h:smax;     %Spot prices
q = T:-k:0;          %Times

V = zeros(N+1,M+1); %Solution matrix initialization
%*************************************************************************

%*************************************************************************
%BOUNDARY CONDITIONS
%
%The boundary conditions should reflect the payoff of the option
%*************************************************************************
%European Option Dirichlet Boundary conditions
V(:,1) = max(s-X,0);
```

```
V(1,:) = 0;
V(N+1,:) = smax - X* exp(r* (q-T));
%************************************************************************

%FD Variables
alpha = 0.5*o^2*s.^2;
beta = r*s;
gamma = -r;

%FIRST STEP WITH ANALYTIC FORMULA
V(:,2) = flipdim(bsmatrix('C', s,100,k,r,r,o),1)';

%Remaining Time steps with Du Fort and Frankel
A = (-beta*k*h)./(h^2 + 2*alpha*k);
B = (2*alpha*k+4*beta*k*h)./(h^2 + 2*alpha*k);
C = (2*gamma*k*h^2-3*beta*k*h)./(h^2 + 2*alpha*k);
D = (2*alpha*k)./(h^2 + 2*alpha*k);
E = (h^2-2*alpha*k)./(h^2 + 2*alpha*k);

A_ = (2*alpha*k)./(h^2 + 2*alpha*k);
B_ = (3*beta*k*h + 2*gamma*k*h^2)./(h^2 + 2*alpha*k);
C_ = (2*alpha*k - 4*beta*k*h)./(h^2 + 2*alpha*k);
D_ = (beta*k*h)./(h^2 + 2*alpha*k);
E_ = (h^2-2*alpha*k)./(h^2 + 2*alpha*k);

Direction = 1;

for i = 3:(M+1) %Time stepping

    for j = 2:N %Spatial stepping
        if (Direction == 1 && j~=N)||(Direction == -1 && j==2)
            V(j,i) = ...
                A(j)*V(j+2,i-1) + ...
```

```
                B(j)*V(j+1,i-1) + ...
                C(j)*V(j,i-1) + ...
                D(j)*V(j-1,i-1) + ...
                E(j)*V(j,i-2);
        elseif ( Direction == -1 && j~=2 )||( Direction == 1 && j==N)
            V(j,i) = ...
                A_(j)*V(j+1,i-1) + ...
                B_(j)*V(j,i-1) + ...
                C_(j)*V(j-1,i-1) + ...
                D_(j)*V(j-2,i-1) + ...
                E_(j)*V(j,i-2);
        end

        Direction = Direction*-1;

    end
end

Premium = interp1(s(1:N+1),V(1:N+1,M+1),S0,'linear');
```

A.2.5 The 2-dimensional Du Fort and Frankel scheme

```
%*************************************************************************
%    TWO DIMENSIONAL DU FORT AND FRANKEL FINITE DIFFERENCE ALGORITHM
%
% Abie Bouwer
%
% November 2007
%
%*************************************************************************

%clear;
```

```
clc;
tic

%INPUT PARAMETERS
S1_0 = 100;
S2_0 = 100;
X = 100;
T = 1.0;        %Time to maturity
r = 0.1;        %Risk free interest rate
o1 = 0.3;       % Volatility of 1st asset
o2 = 0.2;       % Volatility of 2nd asset
p =0.5;         %Correlation

N1 = 50;        %Spatial steps 1st asset
N2 = 50;        %Spatial steps 2nd asset
M = 80;         %Temporal steps

%GRID INITIALIZATION
epsilon = 3;
s1min = X*exp((r-0.5*o1^2)*T-epsilon*o1*sqrt(T));
s1max = X*exp((r-0.5*o1^2)*T+epsilon*o1*sqrt(T));

s2min = X*exp((r-0.5*o2^2)*T-epsilon*o2*sqrt(T));
s2max = X*exp((r-0.5*o2^2)*T+epsilon*o2*sqrt(T));

k = T/M;
h1 = (s1max-s1min)/N1;
h2 = (s2max-s2min)/N2;

s1 = s1min:h1:s1max;
s2 = s2min:h2:s2max;
q = T:-k:0;
[ss1,ss2]=meshgrid(s1,s2);
```

```
V = zeros(N1+1,N2+1,M+1); %Solution matrix  initialization

%BOUNDARY CONDITIONS
V (:,:,1)  = max(max(ss1,ss2)'-X,0);
V (1,:,:)  = U (1,:,:);
V (:,1,:)  = U (:,1,:);
V(:,N2+1,2:(M+1)) = U(:,N2+1,2:(M+1));
V(N1+1,:,2:(M+1)) =U(N1+1,:,2:(M+1));

%ADDITIONAL BOUNDARY CONDITION WITH EXPLICIT SCHEME
V (:,:,2)  = U (:,:,2);
%MAIN SOLUTION
H =  (1./( h1^2*h2^2+h2^2*k*o1^2*ss1.^2+h1^2*k*o2^2*ss2 .^2))';
A = 0.5*H.*(h1*h2*k*p*o1*o2*ss1.*ss2)';
B = H.*(h2^2*k*(o1^2*ss1.^2+r*h1*ss1 ))';
C = H.*(h1^2*k*(o2^2*ss2.^2+r*h2*ss2 ))';
D = H.*(-2*r*h1^2*h2^2*k);
E = H.*(h2^2*k*(o1^2*ss1.^2-r*h1*ss1 ))';
F = H.*(h1^2*k*(o2^2*ss2.^2-r*h2*ss2 ))';
G = H.*(h1^2*h2^2-h2^2*k*o1^2*ss1.^2-h1^2*k*o2^2*ss2.^2)';

for  i  = 3:M+1
    V(2:N1,2:N2,i)  =  ...
        A(2:N1,2:N2).*(V(3:(N1+1),3:(N2+1),i-1)-V(3:(N1+1),1:(N2-1),i-1)-...
            V(1:(N1-1),3:(N2+1),i-1)+V(1:(N1-1),1:(N2-1),i-1))+...
        B(2:N1,2:N2).*V(3:(N1+1),2:N2,i-1)+C(2:N1,2:N2).*V(2:N1,3:(N2+1),i-1)+...
        D(2:N1,2:N2).*V(2:N1,2:N2,i-1)+E(2:N1,2:N2).*V(1:(N1-1),2:N2,i-1)+...
        F(2:N1,2:N2).*V(2:N1,1:(N2-1),i-1)+G(2:N1,2:N2).*V(2:N1,2:N2,i-2);
end
toc
```

A.2.6 The 2-dimensional Du Fort and Frankel scheme without upper and lower boundaries

```
%*************************************************************************
%
%          BOUNDARY FREE 2 DIMENSIONAL DU FORT AND FRANKEL ALGORITHM
%
% Abie Bouwer
%
% November 2007
%
%*************************************************************************

clc;
tic

%INPUT PARAMETERS
s1 = 100;
s2 = 100;
X = 100;
T = 1.0;        %Time to maturity
r = 0.1;        %Risk free interest rate
o1 = 0.3;       % Volatility of 1st asset
o2 = 0.2;       % Volatility of 2nd asset
p =0.5;         %Correlation

M = 100;        %Temporal steps
N1 = 2*M; %Spatial steps 1st asset
N2 = N1;      %Spatial steps 2nd asset
%GRID INITIALIZATION
epsilon = 8;
s1min = log(X*exp((r-0.5*o1^2)*T-epsilon*o1*sqrt(T)));
s2min = log(X*exp((r-0.5*o2^2)*T-epsilon*o2*sqrt(T)));
```

```
s1 = log(s1);
s2 = log(s2);
h1 = (s1-s1min)/M;
h2 = (s2-s2min)/M;
s1max = s1min+h1*N1;
s2max = s2min+h2*N2;
X = log(X);

k = T/M;
s1 = s1min:h1:s1max;
s2 = s2min:h2:s2max;
q = T:-k:0;
[ss1,ss2]=meshgrid(s1,s2);

V = nan*ones(N1+1,N2+1,M+1); %Solution matrix  initialization

%BOUNDARY CONDITIONS
V(:,:,1) = max(max(exp(ss1),exp(ss2))-exp(X),0);

%ADDITIONAL BOUNDARY CODITION WITH EXPLICIT SCHEME
V(2:N1,2:N2,2) = U(2:N1,2:N2,2);

%MAIN SOLUTION
v1 = (r-0.5*o1^2);
v2 = (r-0.5*o2^2);

H = (1./( h1^2*h2^2+h2^2*k*o1^2+h1^2*k*o2^2))';
A = 0.5*H.*(h1*h2*k*p*o1*o2)';
B = H.*(h2^2*k*(o1^2+v1*h1))';
C = H.*(h1^2*k*(o2^2+v2*h2))';
D = H.*(-2*r*h1^2*h2^2*k);
E = H.*(h2^2*k*(o1^2-v1*h1))';
F = H.*(h1^2*k*(o2^2-v2*h2))';
```

```
G = H.*(h1^2*h2^2-h2^2*k*o1^2-h1^2*k*o2^2)';
j=2;
for  i  =  3:M+1
    V((j+1):(N1-j+1),(j+1):(N2-j+1),i) =  ...
        A*(V((j+2):(N1-j+2),(j+2):(N2-j+2),i-1)-V((j+2):(N1-j+2),j:(N2-j),i-1)-...
            V(j:(N1-j),(j+2):(N2-j+2),i-1)+V(j:(N1-j),j:(N2-j),i-1))+...
        B*V((j+2):(N1-j+2),(j+1):(N2-j+1),i-1)+C*V((j+1):(N1-j+1),(j+2):(N2-j+2),i-1)+...
        D*V((j+1):(N1-j+1),(j+1):(N2-j+1),i-1)+E*V((j):(N1-j),(j+1):(N2-j+1),i-1)+...
        F*V((j+1):(N1-j+1),(j):(N2-j),i-1)+G*V((j+1):(N1-j+1),(j+1):(N2-j+1),i-2);
    j=j+1;
end
```

Bibliography

D.-H. Ahn, S. Figlewski, and B. Gao. Pricing discrete barrier options with an adaptive mesh model. *Journal of Derivatives*, 6(4):33–43, 1999.

R. Beneder and T. Vorst. Options on dividend paying stocks. In J. Yong, editor, *Recent developments in mathematical finance*, pages 204–217. World Scientific, May 2001.

T. Björk. *Arbitrage theory in continuous time*. Oxford University Press, second edition, 2004.

F. Black. Fact and fantasy in the use of options. *Financial Analysts Journal*, pages 36–72, Jul/Aug 1975.

F. Black and M. Scholes. The pricng of options and corporate liabilities. *Journal of Political Economy*, (81):637–654, 1973.

M. Bos and S. Vandermark. Finessing fixed dividends. *Risk*, pages 157–158, Sep 2002.

M. J. Brennan and E. S. Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and quantitative Analysis*, 1978.

D. M. Chance, R. Kumar, and D. Rich. European option pricing with discrete stochastic dividends. *Journal of Derivatives*, 9(3):39–45, 2002.

M. Chawla, M. Al-Zanaidi, and D. J. Evans. Generalized trapezoidal formulas for the BlackScholes equation of option pricing. *International Journal of Computer Mathematics*, 80(12):1521–1526, Dec 2003.

M. M. Chawla and D. J. Evans. High-accuracy finite-difference methods for the valuation of options. *International Journal of Computer Mathematics*, 82(9):1157–1165, Sep 2005.

N. A. Chriss. *Black-Scholes and beyond: Option pricing models*. Irwin professional publishing, 1997.

N. Clarke and K. Parrot. Multigrid for American option pricing with stochastic volatility. *Applied Mathematical Finance*, 6(3):177–195, Sep 1999.

L. Clewlow and C. Strickland. *Implementing derivatives models*. Wiley Financial Engineering, 1998.

J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat–conductive type. *Proceedings of the Cambridge Philosophical Society*, 43:50–67, 1947.

E. C. Du Fort and S. P. Frankel. Stability conditions in the numerical treatment of parabolic differential equations. *Mathematical tables and other aids to computation*, 7(43):135–152, Jul 1953.

D. Duffy. Notes on difference equations for the implicit euler, ADE and MADE finite difference schemes. Private communication, 2006a.

D. J. Duffy. *Financial Instrument Pricing Using C++*. John Wiley & Sons, Ltd., 2004a.

D. J. Duffy. A critique of the Crank Nicolson scheme: Strengths and weaknesses for financial instrument pricing. *Wilmott Magazine*, Jul 2004b.

D. J. Duffy. *Finite difference methods in financial engineering*. John Wiley & Sons, Ltd, 2006b.

J. Feldman. Richardson extrapolation. Supplementary notes. Department of Mathematics, University of British Columbia.

S. Figlewski and B. Gao. The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics*, (53):313–351, 1999.

B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, Oct 1988.

V. Frisling. A discrete question. *Risk*, Jan 2002.

J. Gatheral, Y. Epelbaum, J. Han, K. Laud, and O. Lubovitsky. Implementing option–pricing models using software synthesis. *Computing in Science and Engineering*, 1999.

R. Geske and K. Shastri. Valuation by approximation: A comparison of alternative option valuation techniques. *Journal of Financial and Quantitative Analysis*, 20(1): 45–71, Mar 1985.

D. Gottlieb and B. Gustafsson. Generalized du fort-frankel methods for parabolic initial-boundary value problems. *SIAM Journal on Numerical Analysis*, 13(1):129–144, Mar 1976.

E. G. Haug. *The complete guide to option pricing formulas*. McGraw-Hill, 1998.

E. G. Haug, J. Haug, and A. Lewis. Back to basics: a new apporach to the discrete dividend problem. *Wilmott*, Sep 2003.

J. Hull and A. White. Valuing derivative securities using the explicit fintie difference method. *Journal of Financial and Quantitative Analysis*, 25(1):87–100, Mar 1990.

J. C. Hull. *Options, futures & other derivtives*. Prentice Hall, fifth edition, 2003.

E. Isaacson and H. B. Keller. *Analysis of Numerical Methods*. Dover Publications, Inc., New York, 1966.

C. C. Leentvaar and C. W. Oosterlee. American options with discrete dividends solved by highly accurate discretizations. In *Progress in Industrial Mathematics at ECMI 2004*, volume 8 of *Mathematics in Industry*, pages 427–431. Springer Berlin Heidelberg, 2006.

G. Linde, J. Persson, and L. Von Sydow. High-order adaptive space-discretizations for the Black-Scholes equation. Technical Report 2006–021, Uppsala Universitet Information Technology, Apr 2006.

K. A. Lindsay. Numerical solution of partial differential equations, Nov 2005. Drawn from lecture notes by Dr. D.M.Houghton, Department of Mathematics, University of Glasgow.

R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, (3):125–144, 1976.

B. V. Noumerov. A method of extrapolation of perturbations. *Monthly Notices of the Royal Astronomical Society*, 84, 1924.

C. W. Oosterlee, C. C. Leentvaar, and A. A. Vázquez. Pricing options with discrete dividends by high order finite differences and grid stretching. In P. Neittaanmäki, T. Rossi, S. Korotov, E. O. nate, J. Périaux, and D. Knörzer, editors, *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.

J. Persson. Accurate finite difference methods for option pricing. Technical Report urn:nbn:se:uu:diva-7097, Uppsala Universitet, 2006. Digital comprehensive summaries of Uppsala dissertations from the Faculty of Science and Technology 206.

J. Persson and L. Von Sydow. Pricing European multi-asset options using space time adaptive FD-method. *Computing and Visualization in Science*, 10(4):173–183, Dec 2007.

S. Rabinowitz. How to find the square root of a complex number. http://www.mathpropress.com/stan/bibliography/complexSquareRoot.pdf. Reprinted from Mathematics and Informatics Quarterly, 3(1993)54–56.

A. S. Sabau and P. E. Raad. Comparisons of compact and classical finite difference solutions of stiff problems on nonuniform grids. *Computers & Fluids*, 28(3):361–384, Mar 1998.

M. Schroeder. Adapting the binomial model to value options on assets with fixed cash payouts. *Financial Analysts Journal*, pages 54–62, Nov/Dec 1988.

R. Seydel. *Tools for computational finance*. Springer, second edition, 2004.

W. Shaw. Advanced finite difference schemes. Presentation on behalf of Oxford Centre for Computational Finance.

G. D. Smith. *Numerical solution of partial differential equations: Finite Difference Methods*. Oxford applied mathematics and computing science series. Claredon Press Oxford, 3 edition, 1984.

A. Sottoriva and B. Rexhepi. Investigating finite difference methods for option pricing. Technical report, Universiteit van Amsterdam, Jun 2007.

D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method*. John Wiley & Sons, Inc., 2000.

P. Wilmott. *Paul Wilmott on Quantitative Finance*, volume 1. John Wiley & Sons, Ltd., 2000a.

P. Wilmott. *Paul Wilmott on Quantitative Finance*, volume 2. John Wiley & Sons, Ltd., 2000b.

P. Wilmott. *Paul Wilmott introduces quantitative finance*. John Wiley & Sons Ltd., 2001.